



Defence Research and
Development Canada

Recherche et développement
pour la défense Canada



Network-Centric Synthetic Environments: *A Modular Modeling & Simulation/Synthetic Environment (M&S/SE) Framework*

A.L. Vallerand and M. Thompson

Defence R&D Canada – Ottawa

TECHNICAL MEMORANDUM

DRDC Ottawa TM 2004-221

November 2004

Canada

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE NOV 2004		2. REPORT TYPE		3. DATES COVERED -	
4. TITLE AND SUBTITLE Network-Centric Environment: A modular Modeling and Simulation/Synthetic Environment (M&S/SE) framework (U) .				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Defence Research & Development Canada - Ottawa,National Defence Headquarters,Ottawa,CA,K1A OK2				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT see report					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 100	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Network-Centric Synthetic Environments:

*A Modular Modeling & Simulation/Synthetic Environment
(M&S/SE) Framework*

A.L. Vallerand
DRDC Ottawa

M. Thompson
JAMSIM Inc

Defence R&D Canada – Ottawa

Technical Memorandum

DRDC Ottawa TM 2004-221

November 2004

© Her Majesty the Queen as represented by the Minister of National Defence, 2004

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2004

Abstract

A modular Modeling & Simulation/Synthetic Environment (M&S/SE) framework for developing and supporting a network-centric or distributed Collaborative Synthetic Environments (CSE) is proposed and its specific and detailed requirements are documented here. It is proposed that such a framework is specifically designed to promote, foster, augment, and expedite the standardization, interoperability, commonality, reusability, and seamless integration of M&S/SE systems including legacy systems in DND, Other Government Departments (OGD) and beyond. The modular M&S/SE framework relies on a network for communication between the various applications and legacy systems adapted to the framework. To develop and to support a distributed CSE, the M&S/SE framework depends on a layered, functionally separated approach to building dynamically reconfigurable applications. Each layer of the framework provides successive levels of specialization so that as new technology evolves, the implementation of the layer can be easily changed to accommodate new hardware/software or technology changes. Together with the requirements for related and necessary Support Services, this Technical Memorandum documents in a logical approach the Network-Centric M&S/SE framework requirements for an optimally interoperable, common and reusable distributed CSE in DND and beyond, directly supporting Network-Centric Capability Management.

Résumé

On propose un cadre modulaire de la modélisation, de la simulation et des environnements synthétiques (M&S/ES) pour développer et soutenir des environnements synthétiques de collaboration en réseau-central ou distribués (ESC) pour favoriser, stimuler, augmenter, et expédier l'étalonnage, l'interopérabilité, la vulgarisation, la réutilisabilité, et l'intégration des systèmes plus anciens de M&S/ES dans le MDN, d'autres services gouvernementaux (OGD) et autres. Le cadre modulaire de M&S/ES se fonde sur un réseau pour la communication entre les divers applications et systèmes plus anciens adaptés au cadre modulaire. Le cadre de M&S/ES, pour se développer et soutenir un ESC distribué, dépend d'une approche posée et fonctionnellement séparée pour établir des applications dynamiquement reconfigurables. Chaque couche du cadre fournit les niveaux successifs de la spécialisation de sorte que pendant que la nouvelle technologie évolue, l'exécution de la couche puisse être changée pour adapter à de nouveaux changements de matériel ou de technologie. En plus des conditions pour des services relatifs, ce rapport technique documente les conditions Réseau-Centrale de cadre de M&S/ES pour un ESC distribué de façon optimale, interopérable, commun et réutilisable dans le MDN et autre, soutenant directement la gestion Réseau-Centrale de capabilités.

Executive summary

A modular Modeling & Simulation/Synthetic Environment (M&S/SE) framework for developing and supporting a network-centric or distributed Collaborative Synthetic Environments (CSE) is proposed and its specific and detailed requirements are documented here. It is proposed that such a framework is specifically designed to promote, foster, augment, and expedite the standardization, interoperability, commonality, reusability, and seamless integration of M&S/SE systems including legacy systems in DND, Other Government Departments (OGD) and beyond.

Subsequent to the 9/11 events, it has become apparent and necessary that DND be involved in efforts related to improving Canada's capacity to manage and simulate complex/extreme events in a Public Security context. Through a modular M&S/SE framework with its associated Services requirements for developing and supporting distributed CSE, it should allow DND to participate for simulations of Federal, Provincial and Local crisis and the mastering of consequence management systems. Furthermore, it should permit DND to participate, through simulations, into the validation of all governmental authorities (command & control), strategies, plans, policies, procedures, protocols, and synchronized capabilities. The modular M&S/SE framework relies on a network for communication between the various applications and legacy systems adapted to the framework. To develop and to support a distributed CSE, the M&S/SE framework depends on a layered, functionally separated approach to building dynamically reconfigurable applications. Each layer of the framework provides successive levels of specialization so that as new technology evolves, the implementation of the layer can be easily changed to accommodate new hardware/software or technology changes.

Together with the requirements for related and necessary Support Services, this Technical Memorandum documents in a logical approach the Network-Centric M&S/SE framework requirements for an optimally interoperable, common and reusable distributed CSE in DND and beyond, directly supporting Network-Centric Capability Management.

Vallerand, A.L. and M. Thompson. 2004. Network-Centric synthetic Environments: a modular modeling & simulation/synthetic environment (M&S/SE) framework. DRDC Ottawa TM 2004 –221. Defence R&D Canada - Ottawa

Sommaire

On propose un cadre modulaire de la modélisation, de la simulation et des environnements synthétiques (M&S/ES) pour développer et soutenir des environnements synthétiques de collaboration en réseau-central ou distribués (ESC) pour favoriser, stimuler, augmenter, et expédier l'étalonnage, l'interopérabilité, la vulgarisation, la réutilisabilité, et l'intégration des systèmes plus anciens de M&S/ES dans le MDN, d'autres services gouvernementaux (OGD) et autres. Le cadre modulaire de M&S/ES se fonde sur un réseau pour la communication entre les divers applications et systèmes plus anciens adaptés au cadre modulaire. Le cadre de M&S/ES, pour se développer et soutenir un ESC distribué, dépend d'une approche posée et fonctionnellement séparée pour établir des applications dynamiquement reconfigurables. Chaque couche du cadre fournit les niveaux successifs de la spécialisation de sorte que pendant que la nouvelle technologie évolue, l'exécution de la couche puisse être changée pour adapter à de nouveaux changements de matériel ou de technologie.

Sans équivoque, et surtout après les événements du 11 septembre, la demande de partager la simulation normalisée ou standardisée dans des environnements réels, constructifs et virtuels distribués est en forte croissance, alors qu'en même temps, cette demande représente un défi significatif pour les personnes d'organisation et les organismes qui doivent établir le consensus pour réaliser et utiliser ces simulations de préférence normalisées. Tous ces efforts d'étalonnage nécessitent du temps, de l'énergie, et du dialogue pour réaliser le consensus - et dans le monde occupé d'aujourd'hui tous les trois produits mentionnés ci-dessus sont difficiles à obtenir. La présente approche d'un cadre de simulation modulaire a été conçue pour synergiquement lier personnes, processus, modèles et outils de simulation pour permettre l'application de M&S/ES au concept, développement, et expérimentation (CDE), l'essai et l'évaluation (E&E), l'acquisition, l'appui et le support matériel (AM&S), formation, la répétition de mission (M.), et la disposition du système. Il est important de réaliser qu'en l'absence de cadre modulaire d'environnement synthétique, la simulation distribuée, la simulation en condition réseau –Centrale continue d'être difficile et complexe à mettre en place et ensuite à exécuter pour tous les partenaires de simulation.

En plus des conditions pour des services relatifs, ce rapport technique documente donc les conditions Réseau-Centrale de cadre de M&S/ES pour un ESC distribué de façon optimale, interopérable, commun et réutilisable dans le MDN et autre, soutenant directement la gestion Réseau-Centrale de capacités militaires.

Vallerand, A.L. and M. Thompson. 2004. Network-Centric synthetic Environments: a modular modeling & simulation/synthetic environment (M&S/SE) framework. DRDC Ottawa TM 2004 –221. R & D pour la défense Canada - Ottawa.

Table of contents

1.	Introduction	1
2.	Why a Modular M&S/SE Framework?	2
3.	Distributed Collaborative Synthetic Environments (CSE)	4
3.1	Virtual Collaboration.....	5
3.2	Integrated Network-Centric Collaborative Synthetic Environments.....	7
4.	DND Preparedness by Using a Modular M&S/SE Framework	11
5.	Network-Centric Capability Management through a Modular M&S/SE Framework.	14
6.	Key Terms & Concepts and N-Tier Application Architecture	16
6.1	Definitions of Key Terms	16
6.2	Definitions of Key Concepts	16
6.3	N-Tier Application Architecture	19
7.	The Modular M&S/SE Framework Overview	26
7.1	Associated Services Requirements	26
7.2	DND Users Requirements	26
8.	The Modular M&S/SE Framework Description	28
9.	The Modular M&S/SE Framework Segmentation	30
9.1	Framework.....	30
9.2	Simulation Runtime.....	37
9.3	Software Development Environments (SDE).....	44
9.4	Client Applications.....	48
9.5	Server Applications	52
9.6	Distributed HLA Applications	55
9.7	Management Applications.....	56
9.8	Common Synthetic Environment (CSE) Infrastructure.....	59

9.9	Dynamic Synthetic Environment & Computer Generated Forces (DSE/CGF)	62
10.	Services Requirements	73
10.1	Support Services	73
10.2	Educational & Training Services.....	74
10.3	Professional Services.....	77
11.	Conclusion.....	81

List of figures

Figure 1	Integrated View of the decentralized distributed CSE	8
Figure 2	Functional Layers of the CSE Pyramid	9
Figure 3	Synthetic Environment (SE) Pyramid	10
Figure 4	Common Computer System Architecture.....	17
Figure 5	Common Software Program	18
Figure 6	The Seven (7) Layers of Open System Interconnection (OSI) model created by the International Standards Organization (ISO).	19

Acknowledgements

This Technical Memorandum is the product of an ongoing effort to understand, articulate and document the requirements for a modular network-centric Modeling and Simulation & Synthetic Environment framework to develop and support distributed collaborative synthetic environments across DND and GOC. In many ways, it is the result of several requests by clients for such generic documentation. These requirements should prove to be not only instrumental but transformational in developing better system designs, with faster time delivery, with cheaper overall costs or with greater agility, in order to optimize current and future Capabilities. We would like to acknowledge the contributions by Arun Majumdar, VivoMind LLC (visionary and technical conceptual design support) and by Stephane Albert, CAE Inc. (conceptual framework design support). Their contributions, comments and suggestions have greatly enhanced the essence of this Technical Memorandum.

Finally, the authors appreciate the splendid opportunity to write this Technical Memorandum together and the patience of our family, friends, colleagues and clients who saw a bit less of us as a result of this long but fruitful journey.

.

1. Introduction

A modular network-centric Modeling & Simulation/Synthetic Environment (M&S/SE) framework with its associated Services requirements are documented in the present report. This report will demonstrate that such M&S/SE frameworks are crucial in order to develop and support distributed collaborative working environments for commanders, and indeed for all the soldiers, sailors, airmen, and the supporting Operation Research (OR), Scientific and Engineering communities as well as to make it easier to develop common perceptions of the situation and achieve (self-) coordinated responses to situations. There are different types of decisions to be made and therefore different tools and approaches to these decisions are required. The point is this modular M&S/SE framework will be an enabler to give an opportunity to increase speed of command when it is appropriate and it does not force to do so when it is not. Ultimately, this framework would be an enabler to make a command to be relevant, accurate, and timely.

Time is being compressed and, as a result, the tempo of operations is being increased. The cumulative impact of better information, better distribution, and new organizational behaviour provides DND with the capability to create superior value propositions for their decision-makers and operators, and dominate its battlespace. This modular M&S/SE framework should enable in a much easier fashion the testing, conceptualizing and analysis of the distributed operational concepts of net-centric operations, battlefield dominance, precision-guided weapons and related “surgical” engagements, full-dimensional protection, as well as focused future logistics.

2. Why a Modular M&S/SE Framework?

A Modular M&S/SE framework with its associated Services for developing and supporting distributed CSE are important in order to achieve the concept of interoperability, commonality, reusability, and seamless integration. Therefore, this approach would alleviate a burden on DND by deflecting some of these responsibilities to Industry. Furthermore, this approach should allow DND to look for solutions that would swiftly meet its needs, at a cost that it can afford, and guarantee interoperability with other players within Canada and also outside of Canada such as NATO partners. Finally, reusability (interoperability + commonality) would also be accomplished through a structured approach to building a consistent and reusable object library. This reusability could substantially reduce the time and effort needed to develop future integrated simulations.

DND must realize that it is costly and practically impossible to track the use of the M&S/SE Goods & Services throughout the department. Such tracking would avoid unnecessary duplication and redundancy as well as ensure the desired alignment with International open standards. Therefore, DND must come up with innovative ways to overcome this problematic situation. One possible way is for DND to seek to get as many as possible applications that are already integrated to a simulation platform from either the vendor and/or its associated value-added partners to meet its solutions as long as the platform would also allow for third parties to integrate their products; thereby *stimulating* (not stifling) follow-on business in Canada from smaller value-added companies.

The integration is the key value in terms of the current best practices. Decades of hard lessons learned have shown that an ad-hoc mixture of interconnected services and components usually fails to work. Even in such established areas, such as CORBA or XML, there are new standards arising every week. DND should not want to engage in becoming an Information Technology (IT) shop just to be able to “glue” or try to glue all the pieces together.

This modular M&S/SE framework could allow DND to do the following:

- a. Capitalize on investment in various technology development areas i.e. provide the “glue” to put different models, simulations and players together into a distributed collaborative synthetic environment;
- b. Integrate capabilities from existing programs and merge capabilities for overarching goal;
- c. Leverage legacy technology base by using adapters and plug-ins and using new technologies to extract the intellectual capital from obsolete legacy systems;
- d. Leverage the DND funded initiatives;
- e. Identify synergistic intra-division relationships within DND;
- f. Identify synergistic inter-directorate relationships within DND;
- g. Consistent Battlespace Picture (CBP) for Canada or its allies;
- h. Configurable Command Center (CCC);
- i. Dynamic Command and Control (DC2) for:
 - i. Force Support (Logistics);
 - ii. Force Application (Shooter);
 - iii. Force Enhancement (Intelligence, Surveillance & Reconnaissance) ;
 - iv. Air Superiority (Defensive Air);

- v. Maritime Superiority (Defensive Land/Sea)
- vi. Develop new C2 technologies that are highly flexible in an info-centric environment (Infostructure); and
- vii. Demonstrate new C2 capabilities through a Series of critical technology integration experiments designed with operational performance metrics.
- j. Global Grid interconnectivity for coalition development;
- k. Defensive Information Warfare doctrine development; and
- l. Understand how effects based operations change with differences in the order of battle using these new approaches.

Unequivocally, the demand for sharing standardized simulation into distributed live, constructive and virtual environments is strong, growing, while at the same time, organizing people and organizations to build consensus for achieving and using those standards remains a significant challenge. Standardization efforts needs time, energy, and dialogue for achieving consensus – and in the busy world of today all aforementioned three commodities are in short supply.

This approach synergistically would link the people, processes, and simulation models and tools to enable the application of M&S/SE to Concept, Development, and Experimentation (CDE), Test & Evaluation (T&E), Requirements, Material Acquisition & Support (MA&S), Training, Mission Rehearsal (MR), and Disposal projects.

3. Distributed Collaborative Synthetic Environments (CSE)

A modular M&S/SE framework with its associated Services for developing and supporting distributed collaboration synthetic environments (CSE) would allow the creation of Virtual organizations that would bring the necessary people and processes together to accomplish a particular task. When the task is over, these resources would be returned to other tasks. Virtual organizations, enabled by networking, would allow DND to take advantage of the potential gains in productivity that are associated with virtual collaboration, virtual integration, and outsourcing. These individuals can be geographically dispersed. One of the major payoffs of distributed CSE will be in an improved product design process – measured by any one of the following metrics: faster schedule, less costly, overall better system design or a more agile system design.

However, there is no guarantee that simply hooking things up across the battlespace and collaborating together without appropriate organizational and doctrinal changes will increase warfighting effectiveness. In fact, there is every possibility that the unintended consequences of wiring up the battlespace, collaborating together, and “hoping for the best” will, in fact, degrade performance particularly if doctrine, organization, training, support services, and other key elements of the process are not changed to take advantage of the new configuration. Therefore, the road to network centric warfare based upon distributed CSE needs to be richly populated with analyses, experiments and “use cases” in order to understand how DND can reap the huge potential of distributed CSE, while avoiding some of the pitfalls.

Transforming distributed CSE from a concept into a reality will require that the people’s roles, responsibilities, tasks, decisions, connectivity i.e. links among them, and the nature of the information and products that are exchanged i.e. the degree of coupling be defined, work out, and implemented collectively in unison.

It is unlikely that the proper degree of coupling i.e. positive distributed collaborative effects can be realized without having a relatively high-performance communications capability and computational capability [such as the M&S Resource Repository Net i.e.: MSRRNet] providing access to appropriate information and model sources, and allowing seamless interactions among entities in a “plug and play” fashion¹. This is called the “infostructure.” This sort of picture implies that somehow all of the sensors are actually linked together. While this makes sense conceptually, it may not make sense in practice. The fact that actors (shooters) do not inherently own sensors, and decision makers do not inherently own actors, whereas currently in *platform-centric* operations they own weapons and weapons have their own organic sensors, will require more in-depth investigations.

¹ This document hopes to reveal to the reader that the “plug-and-play” concept of today’s M&S has been overused and abused in recent years. In fact, without a modular open architecture framework, or without all players using the same SE tool, “plugging” new or legacy entities requires in many cases significant non-trivial, specialized software integration work before any “playing” takes place, to the surprise of many who had been led to believe or were happy to believe otherwise. In a closed architecture, such integration work usually can only be performed by the Industry who delivered that closed architecture (i.e.: “black box”) in the first place, hence the current push toward open architecture, shared by many.

Distributed CSE should give the opportunity to explore the vast middle ground between the top-down hierarchical command and control approach and the highly decentralized model of small sections assigned pieces of the problem with only their organic capabilities, in a bottom-up reactivity to top-down directive. This vast middle ground should allow considering a host of command and control approaches, many of which could be used simultaneously in the battlespace of the future, each optimized for a specific task or function. The overall design of command and control, the way each mission, function, and task will be managed, will need to be conceived in such a way as to bind the overall behaviour of the forces. Adoption of distributed CSE should provide DND with the ability to enlarge the engagement envelope, reduce risk profiles, increase analyzing, experimenting and operating tempo and responsiveness. Distributed CSE, through a set of tightly coupled processes, will:

- a. Facilitates an understanding of emerging capabilities;
- b. Fosters innovative concepts;
- c. Expedites the testing and refinement of these concepts; and
- d. Focuses efforts on the development and deployment of coherent Mission Capability Packages (MCP).

In order to satisfy the needs and mandate of DND, distributed CSE will need to be more than skin-deep. It will need to be built-in from the bottom up, so that the best way to accomplish an exercise, experiment, or task, given the available information and assets, can be employed. Regrettably, there are significant institutional barriers to achieve distributed CSE. To maximize the chances of success, it is required to foster true collaborations in the process of co-evolution, investment strategy, and education and training efforts. First, the introduction of technology in the form of a system, or set of materials, is no longer the focus or objective. Rather, the objective is to support a set of MCP, in a system-of-systems approach. Second, adequate emphasis needs to be placed on MCP being born Joint; otherwise it is likely that stove-piped MCPs will be produced. Third, co-evolution is a process of discovery and testing. The answer will not be known in advance. Thus, the process needs to be devoid of the pass/fail mentality common today. Fourth, the heart of the co-evolution process is experimentation, neither demonstrations nor exercises, although there is a role, albeit a reduced one, for both of these in the process. Fifth, the process is iterative or spiral in development.

3.1 Virtual Collaboration

Distributed CSE with autonomous agents support, called Virtual collaboration, goes far beyond simple sharing of information. It enables elements of the warfighting ecosystem to efficiently and effectively interact and collaborate in the virtual domain, moving information instead of moving people and achieving a critical knowledge mass. Key component technologies such as video teleconferencing (VTC), virtual whiteboards, collaborative engineering processes and tools, distributed simulation software, autonomous agents, and collaborative planning and managing applications enable virtual collaboration.

Intelligent agents for learning methodology applications [in the context of distributed mission training (DMT)] should be able to determine:

- a. The state of the exercise;
- b. The learning objectives; and
- c. Trainee process being made toward achieving the learning objectives.

Intelligent agents, language parsers, and instructor surrogates are tools categories that make use of the computational capability of software to search for pre-defined relationships in data collected in real-time. These tools can be used for real-time feedback, student remediation and post exercise analysis.

A distributed CSE could be employing cognitive agents or autonomous software components, which are able to identify behaviours of a specific type, analyze them, and implement remediation or annotation for After Action Review (AAR) during all phases of the event. These Artificial Intelligent Instructors (AII) tools will execute independently, providing instructor support and objective analysis through behavioural and cognitive modeling. AII's will manipulate other software systems and tools such as query databases and send messages as needed to analyze and remediate student performance. AII will have the capability to employ formats that are specific to the learning objectives and level of training of the participants. Examples of this include early stage training in which intelligent tutors offer hints or correction during execution of events, to mission rehearsal scenarios that only offer remediation when activated by a human instructor or provided in debrief/AAR.

The distributed CSE should bring about changes that will ultimately merge management, planning, and execution into an integrated, dynamic adaptive progress. This will require effective interactions between not only decision entities and actors, but also sensors. The support of autonomous agents for distributed CSE may take the form of any one of several automated decision or information processes, including decision aids, expert systems, trained neural nets, or genetic algorithms, each autonomously performing selected tasks for distributed CSE entities.

Looking forward and considering trends in allied countries as well as the growth of wireless and distributed technologies, service technologies [the internet] and knowledge-based technologies, brings about the issue of business infrastructure forces as follows:

- a. Mobility;
- b. Context;
- c. Knowledge Representation (KR); and
- d. Agent systems.

Mobility is what is called “ubiquitous computing” and this imposes unique requirements on interoperability, the key to which is “context”. A trivial example of context effects is how location and perhaps language preference *adapts* service *delivery* to individuals working together but speaking different languages. Different context-aware infrastructures and devices have to co-exist and be interoperable.

To support interoperability, a context-sensitive model of *information is needed* providing for how context can be *wrapped and accessed* at different levels of roles, abstraction, security and detail to both user and purposes at hand. KR provides the means and the *active* personalization for mobile users. Semantic Web, for navigation, and Conceptual Graphs, for persistence, are key enablers e.g. a *knowledge context* wraps all services, their descriptions, and user specific personalization information (XML). Although, *context dependencies* become important the major advantage is this approach is fully abstract, easily distributed and needs no central configuration and control.

An agent-based architecture is characterized by combining services, mechanisms and business processes into ‘active’ entities that resist failure, adapt and act pro-actively and autonomously.

The agents respond to the demand in their environment by composing business processes providing services.

Agent-Oriented is emerging as a new paradigm in software and information systems engineering and is based on modeling some facets of how a human would do the job with some intelligent decision making built-in, in order to provide a more “personalized” or adaptive capability to an end-user. It offers high-level abstractions that facilitate the conceptual and technical integration of communication and interaction with information systems and compliance with the physical and social dynamics of interacting individuals and institutions.

3.2 Integrated Network-Centric Collaborative Synthetic Environments

DND will rely increasingly on technology as the force-multiplier. To increase force-multiplication, and hence, combat power, DND will need an integrative approach to technology, warfare, and military organizations. A distributed CSE that permits strategic, tactical and asymmetric force-multiplication is needed in order for Canada to defend and participate effectively and decisively in military operations at home and in coalitions abroad.

An underpinning factor of the distributed CSE is that only an integrated and all-encompassing approach integrating the three pillars of DRDC [i.e. a) R&D and S&T awareness, b) a network-centric simulation environment, like the JSimNet, and c) accessible sensors and source repository network, like the MSSRNet], will dramatically reduce cost, multiply value and augment the speed, tempo and agility of high quality support to operational decision making

In order to set the context to understand why a technological future and specifically why an integrated decentralized CSE is crucial to Canadian military effectiveness, history provides us the answer: In “Megatrends”, John Naisbitt, writes about technology itself without knowing that a “Revolution in Military Affairs” did not exist yet:

“There are three stages of technological development. During the first stage, technology takes the path of least resistance, that is, it is applied in ways that do not threaten people. Second, the technology is used to improve previous technologies e.g. Today’s word processor is nothing more than an improved typewriter. In the third stage, new directions of uses are discovered that grow out of the technology itself. New information technologies gradually give birth to new activities, processes, and products”. [Naisbitt, 1982]

Information Technology (IT) has now entered the third stage of technological development giving rise to new systems, processes, and opportunities. The list of new activities based on IT enables waging war based on the use of information and new terms like “infostructure”, network-centric-warfare (NCW) or network enabled capability (NEC), integrative warfare, Computers, Communications Command and Control, Intelligence, Surveillance, and Reconnaissance (C4ISR), and “system of systems”, are all critically created and defined by information technology.

Strategists such as Sun Tzu had spoken of the importance of knowing the enemy and one’s self (training, readiness, and collaborative forces) centuries ago in order to defeat an opponent’s strategy before battle (Giles, 2003). Today, the methods for entering the warfare decision-making cycle and gaining insights into strategy and military capabilities are powered by information

technologies such as simulation capability (i.e.: the JSimNet) and repository environment (i.e.: DND SECO's MSRRNet). Synthetic Environments (SE) are just one piece.

First, the entry fee is a high-performance R&D data grid or network [which corresponds to the information part] that provides an infrastructure for future computing and communications from DND/DRDC efforts. Knowledge-based sensing networks [which corresponds to the MSRRNet] rapidly generate high levels of battle-space awareness and synchronize awareness with military decision-making activities. The JSimNet could enable the operational architectures of diverse elements between military and civilian systems to act as interoperable engagement systems. Engagement systems [which correspond to the action part in warfare and simulations] exploit and convert this interoperability into awareness and translate this capability into force-multiplied effects, at either a combat level or in a civilian disaster management level.

Without the integrated understanding that can only emerge with an integrative strategy, it is impossible to determine what is an “available capability”, what are total and realistic cost control at all levels, or what is the capability to impact force in asymmetric situations. The integrated network-centric CSE will become the core foundation of situational training, combat power, as well as strategic capabilities such sustainment, force protection/generation at the military and civilian levels.

The Holistic View is Integrative

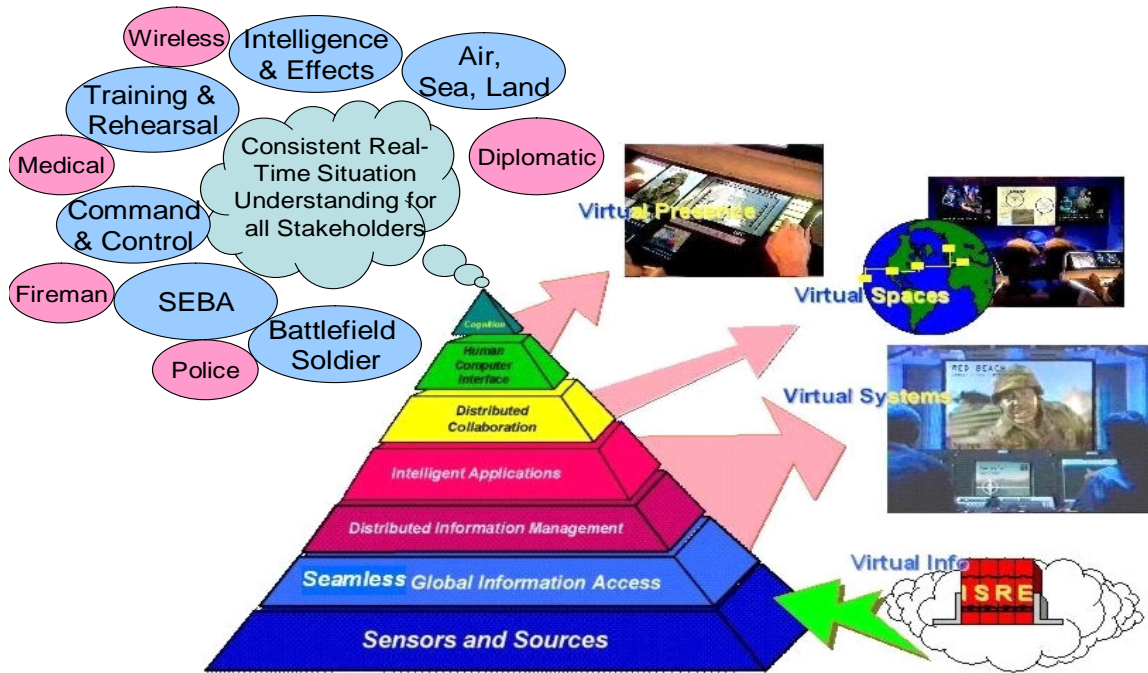


Figure 1 Integrated View of the decentralized distributed CSE.

As presented in [Figure 1](#) above and [Figure 2](#) below, Sensors and Sources are the core repositories or acquired information as objects that provide data or information to other higher order processes. Seamless global information access whether through wireless, telephone, networks or fibre optics need to be managed both at points of origin and at locations of need, which are

distributed in time and geography. The total availability of anywhere on-demand information means that information has been “virtualized” from its physical form in one location or place of storage.

However, distributed collaboration needs services that are adaptive and these can only be rendered by applications that have basic decision-making intelligence e.g. ensuring that the consumer of information gets it immediately in the native language of choice (multilingualism). Human computer interfaces are remote but integrated collaboration can be unified in a distributed CSE that brings together all the core layers into a working system of systems.

The essence of Future Forces lies in understanding how to optimize the cognitive domain, from the point of view of planning, thinking, or figuring out how to coordinate and optimize the interactions between all accessible resources and capabilities in a way that the customized force can yield just the right level of effect whether in combat power for DND or in disaster management with our public security partners.

FUNCTIONAL LAYERS PYRAMID

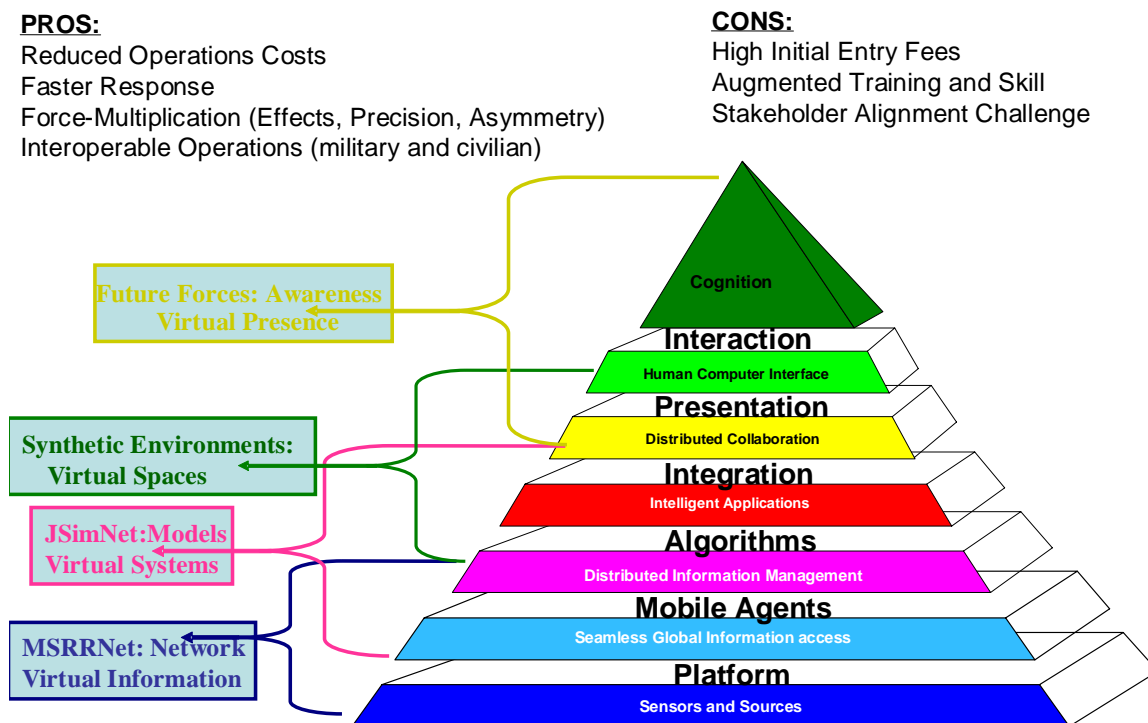


Figure 2 Functional Layers of the CSE Pyramid.

The Synthetic Environment (SE) pyramid as shown in [Figure 3](#) below can be explained as follows:

Degree of Reality: The convergence of the Problem Space with the Implementation Space is the degree of reality of the *Model*. At the top of the pyramid, in the ideal limit, the behaviours in the SE can exactly predict or match the behaviours in reality!

Knowledge Representation: It is the characteristics of the structure of data and information in a generalized form so that it can be applied to different circumstances.

Model: A representation of “what’s important”. It emphasizes *concepts* and *relationships* relevant to the reality (*variable resolution*), hides unnecessary details (*variable fidelity*), and focuses on elucidating structure and/or function, role, and purpose. Models are (can be) precise, unambiguous, complete, executable, and verifiable between Problem and Solution domain and the Real World itself.

Problem Space: This means that sufficient *knowledge* is at hand to define the problem and to have the means to solve it. The description of a problem does not need any technology at this stage. Therefore it is independent of any particular form implementation i.e. implementation-independent.

Solution Space: This is, however, very *dependent on the implementation*. Many problems cannot be solved in a reasonable amount of time without the right algorithms or special purpose processes and networks. In order to solve a problem in the implementation space, it has to be *mapped* from the problem space onto algorithms, networks and hardware, to be solved.

The Synthetic Environment (SE) Pyramid

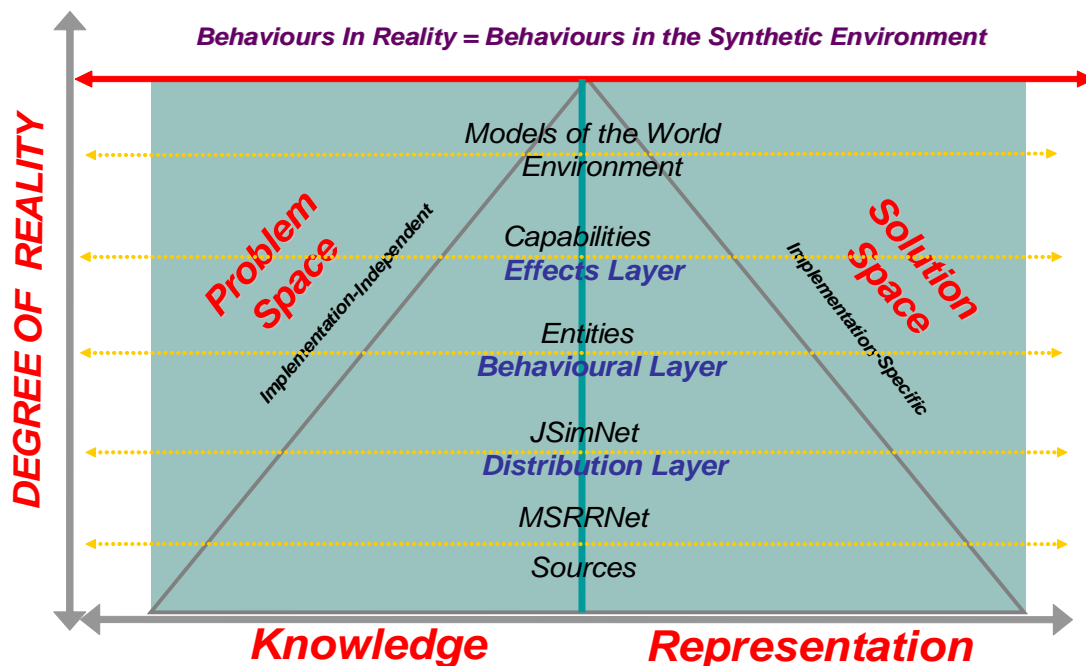


Figure 3 Synthetic Environment (SE) Pyramid.

4. DND Preparedness by Using a Modular M&S/SE Framework

There is an eminent need to maximize efficiency, effectiveness, reusability, interoperability, and return on investment (ROI) of the application of simulation models and tools to obtain high-fidelity-capable, standardized, distributed, interactive and integrated synthetic environments scenarios generation into DND in order to encourage and leverage the coordination and costs associated with conducting cross-pollinated multifaceted real-life scenarios among its department, other departments, and also across the different levels of government.

Thus far, DND has not been able to model itself or even the country as a national system-of-systems (in terms of national capabilities) and therefore DND's knowledge is very limited in terms of how it should optimally address a crisis and what capabilities ought to be delivered to that crisis. As per example, how well answered are the following questions asked for preparedness and reaction's optimization:

- a. What are the possible scenarios?
- b. What would really be possible or what could happen under various circumstances? and
- c. How much useful work can it be done collectively prior to the crisis?

The aforementioned questions would be better answered and addressed through a distributed Collaborative Synthetic Environments (CSE).

For DND, a key point is that there is none or little cross-functional planning or modeling systems to identify critically weak areas in an information-age conflict. DND has few elements of an effective counter-terrorism solution that can identify and propose a safer world. This includes modeling and simulation of a much broader array of data than current systems are capable of doing, discovering information and knowledge from the data and applying intelligence to cost-saving with effective safety increases, creating models of scenarios, and analyzing these systems in a distributed collaborative synthetic environment to determine the most probable current or future scenario. Cross-functional planning or modeling systems would allow disparate applications, DND divisions, systems and dynamic models to be integrated in a manner that minimizes (or eliminates) the need to rework or recreate the system as new models or changes would be added to DND (synthetic models).

In a climate of complex asymmetric threats, DND does not have access to system-of-systems modeling environments in support for Canada "Public Safety". DND has fundamental interoperation issues that have not yet been modeled such as:

- a. How would military and local government address and respond to a catastrophe?
- b. What are the interoperation barriers and what can be done about them collectively prior to the next 9/11 event?

The M&S/SE system that could support difficult and otherwise costly effort would provide acritical path analysis. DND is ill-equipped to model the network-centric problems of interoperation, security, information or knowledge management-based situations of conflict or breakdown without an innovative modular M&S/SE framework with its associated Services.

The general context for a DND [and perhaps eventually nationally] recommended modular M&S/SE framework would identify strategies to solve some critical action areas for DND [and Canada]. DND decision-makers face an ever-increasing challenge to balance maximum flexibility for the mission with a multitude of other resources use that must address the social, political, military and economic goals due to the special characteristics of Canada itself (smaller workforce compared to US, less accessible monetary supply, great natural resources, wide geography, etc.). In addition, these goals encompass environmental requirements for maintaining Canada's health and sustainability over the long term. To meet these challenges, synthetic environments deployed within a DND- [and perhaps eventually nationally] mandated modular M&S/SE framework with its associated Services initiative is needed to interoperate geographic information systems (GIS), remote sensing applications, scientific visualization, and decision analysis techniques that are fast becoming tools of the trade. However, when used individually by different DND activities as they are today, these tools are often limited because they evaluate potential impacts for only one particular model or system characteristic at a time, while holding the remainder of the system static. The result means that DND is partially "blind" to hidden problems that may arise in the future. Homeland Defence Security in the US (<http://www.whitehouse.gov/homeland/>) as well as the Federal Ministry Public Safety and Emergency Preparedness (PSEP; <http://www.psepc-sppcc.gc.ca/>), the Provincial Ministry of Community Safety <http://www.mpss.jus.gov.on.ca>), the Regional Municipality of Ottawa-Carleton Emergency and Protective Services (<http://www.rmoc.on.ca/>) were recently created to alleviate that deficiency, but interoperable simulation capabilities are not yet available.

These hidden problems of tomorrow can be identified now with a new M&S/SE strategy for security, health and prosperity as follows:

- a. *Dynamic Environments* – Changing political interactions, military, urban, paramilitary and peace-keeping as well as within Canada (provinces, local government and local service mobilization) i.e. characterizing risks to peace for Canadian security interests;
- b. *Distributed Cognitive Activity* – People becoming smarter and aware as they engage within the systems and in terms of leveraging its effects;
- c. *Complex Human-Machine Interaction* – New opportunities to identify improvements in DND's education and training policies at all levels;
- d. *Complex Human-Information Interaction* – A new and uncharted area for DND to provide knowledge it does not yet have of this area and which may provide critical value against threats to Canada's security;
- e. *Network-Centric Effects Based Operations* at all levels for Canadian Superiority in intelligence, information-based-analytics and execution;
- f. *Enhanced National Security* without violation of privacies;
- g. *Human Centered emphasis* – Using simulation with Human Behaviour Representation (HBR) provides a new perspective on business structures and processes for human work optimization; and
- h. *State Space Geographic Planning and Utilization* for any scenario at any time – Enhanced security for Canada.

Modeling and simulation in support of adaptive system/environment of Canada's natural resources, geographic features, capabilities management, emergency preparedness and effects of change management can be better accomplished through a dynamic, integrated, and flexible

approach that incorporates scientific and technological components into a comprehensive modular M&S/SE framework that:

- a. Focus more on policy and economic issues;
- b. Study and address issues using approaches that may be currently impractical; and
- c. Work at a higher conceptual level when integrating and interchanging models,

resulting into the following:

- a. Improved situational awareness;
- b. Improved knowledge awareness;
- c. Improved political and coalition understanding;
- d. Decrease re-planning response time;
- e. Provide accurate asset tracking;
- f. Evaluate greater number of plan options; and
- g. Provide flexibility in dynamic situations with interleaved planning and execution.

5. Network-Centric Capability Management through a Modular M&S/SE Framework

DND uses applications [systems] to provide services [capabilities], but over time, it becomes difficult integrating them. In the Industry this is known as the EAI (Enterprise Application Integration) problem. Furthermore, as these systems become “obsolete” they often continue to be used in parts of newer business processes. This leads to the “legacy problem” and to a loss of efficiency as these systems are forced to awkwardly emulate newer business processes. Therefore, integrating heterogeneous systems remains a immense challenge because systems were never built with the intention of becoming integrated with the rest of the organization. The result is a hodgepodge of systems that fails to communicate effectively with each other and in consequence slow down the pace of military business delivery.

The essence is to design systems from the ground up to be service providers and to render obsolescence itself “obsolete”. An architecture is required that can handle the increasing complexity of systems by real-time lookup and dynamic binding of resources to provide business services. Its right design and use is the key to availability, failure recovery, quality of service and ubiquity.

By applying the modular M&S/SE framework, an architect creates the *conceptual level* (“what”) of the requirements and designs the *logical level mechanisms* (“how”) (according to which business processes should form systems) and the *physical level context*, by which means, the resulting capability is provided. Hence, the modular M&S/SE framework can be characterized by the combination of *conceptual*, *logical* and *physical*.

Designing DND systems depends on decomposition of these high-level abstractions to levels of detail for both organizational “engineering” (business process re-engineering) and application “engineering” (“make” or “buy” applications). Many modern methodologies overly focus on component design where a process is seen as a stable set of components that interact to provide some functionality. These processes are later combined to form a system that provides the capability.

In the modular M&S/SE framework, it is the capabilities that are composed that drive architecture of the components and not the other way around. In the DND world, there is a growing recognition and need for the assembly of capabilities on demand. However, the architect must address issues with respect to capabilities structures before the design phase begins as follows:

- a. Security and trust between capabilities;
- b. Creating shared contexts between multiple data and information repositories;
- c. Distribution of interactions between users in different locations playing different roles in the organization;
- d. Coordination and control of the overall processes and workflows;
- e. Coordination of business capabilities along various lines of businesses (for example, DND may have overlaps between Unmanned/Uninhabited Vehicle Systems (UVS) for the Air Force as a line of business and UVS for the Army, leading to the concept of “support” as the coordination point between Air Force and Army lines of business from the capability perspective); and

- f. Understanding communication needs to facilitate collaboration in organization applications and mobile environments.

At the design phase, the immediacy of capabilities-on-demand poses additional challenges as follows:

- a. Services that fulfill the essence of the information needs of people or software agents in the business environment;
- b. Infrastructure or support services;
- c. Link-Layer Services. Examples are connectivity services, delivery channel services, event handling services, transformation services, etc... This type of services is delivered by middleware;
- d. Technical services - that give access to personalized use of 'hard' technology components like PDAs, databases and networks;
- e. Reduction of cost caused by platform dependency - Business applications should not directly interact with technical applications. Changes in the technical infrastructure should have no effect on business applications or vice versa;
- f. Reduction of business application development time and cost caused by re-programming infrastructure services. One solution for infrastructure services like connectivity, routing, and security requires an initial investment but can save much time and money in future application development; and
- g. Agent-based business solution scenario - Agents can be strategic or opportunistic. A strategic agent plans and an opportunistic agent seeks opportunities as they emerge. The most important factors areas follows:
 - i. Fast time-to-operations;
 - ii. Autonomous configurations and implementation;
 - iii. Work-on-demand; and
 - iv. Adherence to operators' requirements or standards.

6. Key Terms & Concepts and N-Tier Application Architecture

Prior to present and discuss the modular M&S/SE framework with its associated Services requirements, it is deemed necessary to provide the definitions of key terms and concepts, and application architecture in the context of software. There is no standard, universally-accepted definition or concept for many key terms related to software as it is a field in its infancy, although their roots run deep in software engineering. While there is no standard definition or concept, there is also no shortage of them.

6.1 Definitions of Key Terms

The definitions for key terms in the context of software are as follows:

- a. *Architecture* – It specifies at an abstract level, free of implementation or design details, the group of harmoniously related modules that work together coherently to provide a timeless view of a system (or system of systems) that provides a complete answer to a set of desired requirements (features, functions or any other requirements);
- b. *Framework* – It instantiates the architecture. It is the geometry of a set of modules, often arranged geometrically in layers (but other geometries are used sometimes), of which the modules are components or sets of components of related functionality. This is why software engineers will usually say "component framework". A framework is a refinement of architecture and infrastructure because it is as close to an application domain (e.g. M&S, Financial Services, Avionics, etc...) as is possible *without* design or implementation;
- c. *Infrastructure* - It is a group of modules or a layer or a set of components that together provide fundamental support to an architecture. An infrastructure is a necessary prerequisite for architectures that provide expertise because infrastructure define the internal architecture for an abstract domain (e.g. a software architecture with a knowledge management infrastructure permits intelligent architectures and, therefore, intelligent applications to be built);
- d. *Design* - This is the internal process view of a component and involves algorithms and procedures (usually); and
- e. *Implementation* - This is usually the choice of language in which to write an algorithm for a computer to execute.

6.2 Definitions of Key Concepts

The definitions for key concepts in the context of software are as follows:

An organized list of instructions that, when executed, causes the computer to behave in a predetermined manner is called a *program*. Without programs, computers are useless. A program is like a recipe. It contains a list of ingredients (called variables) and a list of directions (called

statements) that tell the computer what to do with the variables. The variables can represent numeric data, text, or graphical images.

There are many programming languages – C, C++, JAVA, Pascal, BASIC, FORTRAN, COBOL, PROLOG, and LISP are just a few. These are all high-level languages. One can also write programs in low-level languages called assembly languages, although this is more difficult. Low-level languages are closer to the language used by a computer, while high-level languages are closer to human languages.

Eventually, every program must be translated into a machine language that the computer can understand. This translation is performed by *compilers*, *interpreters*, and *assemblers*.

When software is bought, it is normally bought as an executable version of a program. This means that the program is already in machine language - it has already been compiled and assembled and is ready to execute.

A program or group of programs designed for end users. Software can be divided into two general classes: *Systems* software and *Applications* software (see [Figure 4](#) below). Systems software consists of low-level programs that interact with the computer at a very basic level. This includes such as the operating systems, compilers, and utilities for managing computer resources. In contrast, applications software (also called *end-user programs*) includes such as database programs, word processors, and spreadsheets. Figuratively speaking, applications software sits on top of systems software because it is unable to run [execute a program] without the operating system and system utilities.

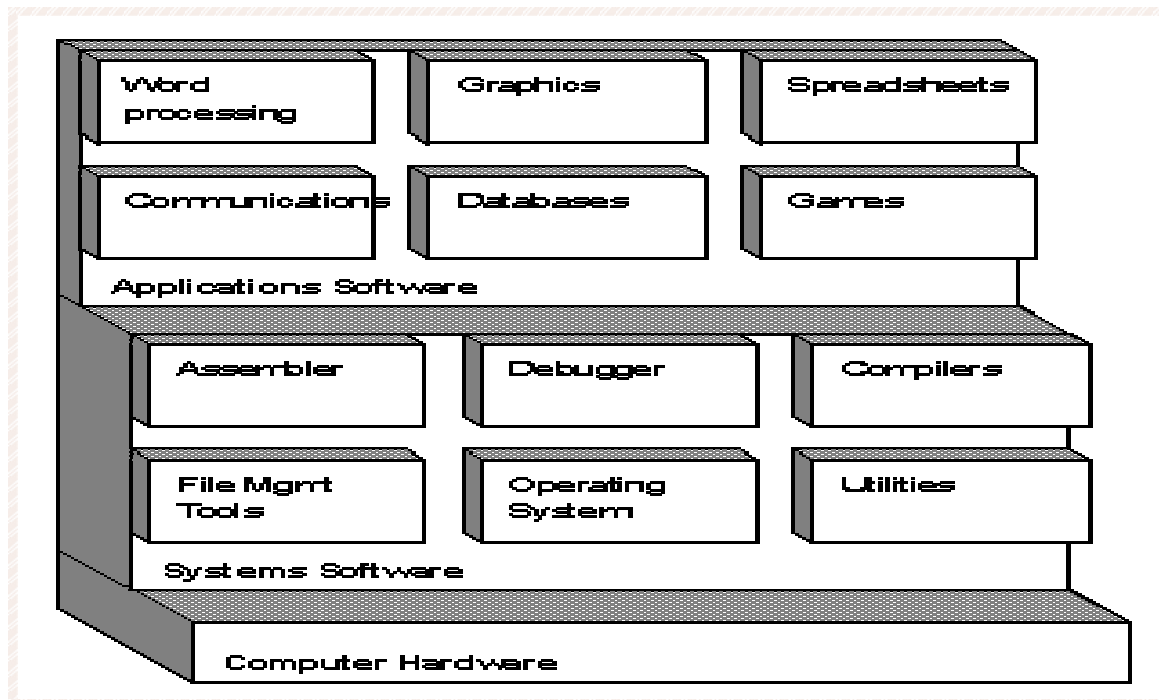


Figure 4 Common Computer System Architecture.

The *operating system* is the most important program that runs on a computer. Every general-purpose computer must have an operating system to run other programs. For large systems, the operating system has even greater responsibilities and powers. It is like a traffic cop - it makes

sure different programs and users running at the same time do not interfere with each other. The operating system is also responsible for security, ensuring that unauthorized users do not access the system.

The Operating Systems (OS) can be classified as follows:

- a. Multi-users allow two or more users to run programs at the same time. Some operating systems permit hundreds or even thousands of concurrent users;
- b. Multi-processing supports running a program on more than one CPU;
- c. Multi-tasking allows more than one program to run concurrently;
- d. Multi-threading allows different parts of a single program to run concurrently; and
- e. Real-time responds to input instantly. General-purpose operating systems, such as Windows and UNIX, are not real-time. *Real time* can also refer to events simulated by a computer at the same speed that they would occur in real life.

A *compiler* is a program that translates source code into object code (See [Figure 5](#) below). The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the instructions. Thus, a compiler differs from an *interpreter*, which analyzes and executes each line of source code in succession, without looking at the entire program. The advantage of interpreters is that they can execute a program immediately. Compilers require some time before an executable program emerges. However, programs produced by compilers run much faster than the same programs executed by an interpreter. Every high-level programming language (except strictly interpretive languages) comes with a compiler. In effect, the compiler is the language, because it defines which instructions are acceptable. Because compilers translate source code into object code, which is unique for each type of computer, many compilers are available for the same language. For example, there is a C++ compiler for PCs and another for LINUX. In addition, the compiler industry is quite competitive, so there are actually many compilers for each language on each type of computer. More than a dozen companies develop and sell C compilers for the PCs.

Utilities differ from applications mostly in terms of size, complexity and function.

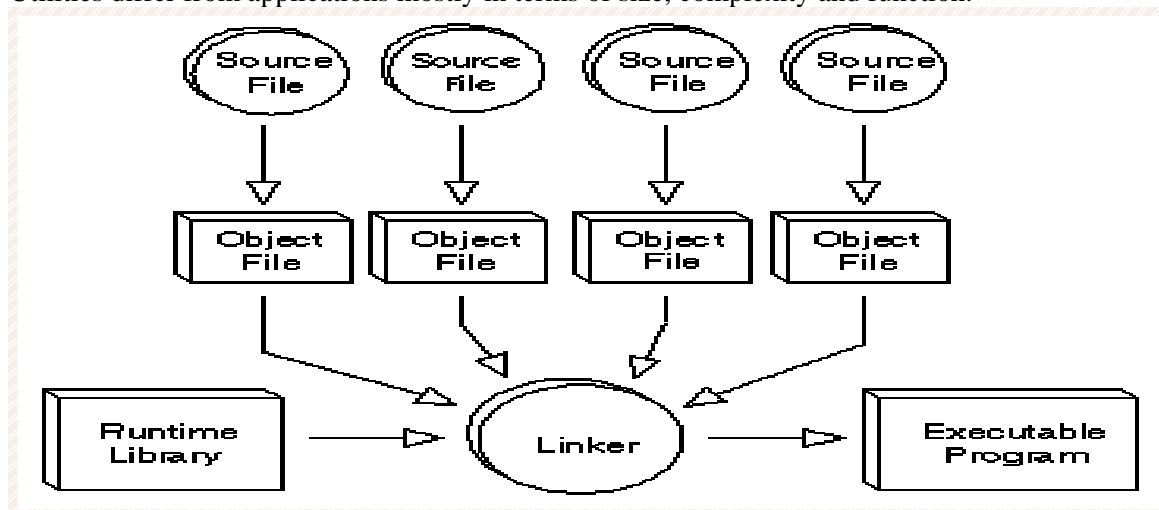


Figure 5 Common Software Program.

6.3 N-Tier Application Architecture

N-tier application architecture provides a model for developers to create a flexible and reusable application. By breaking up an application into tiers, developers only have to modify or add a specific layer, rather than have to rewrite the entire application over, if they decide to change technologies or scale up. In the term "N-tier," "N" implies any number - like 2-tier, or 4-tier; basically, any number of distinct tiers used in your architecture. Application architectures are part of Layer 7 of the Open Systems Interconnection (OSI) Model. (See [Figure 6](#) below)

In 1983, the International Standards Organization (ISO) created the OSI, or X.200, model. It is a multilayered model for facilitating the transfer of information on a network. The OSI model is made up of seven layers, with each layer providing a distinct network service. By segmenting the tasks that each layer performs, it is possible to change one of the layers with little or no impact on the others. For example, you can now change your network configuration without having to change your application or your presentation layer.

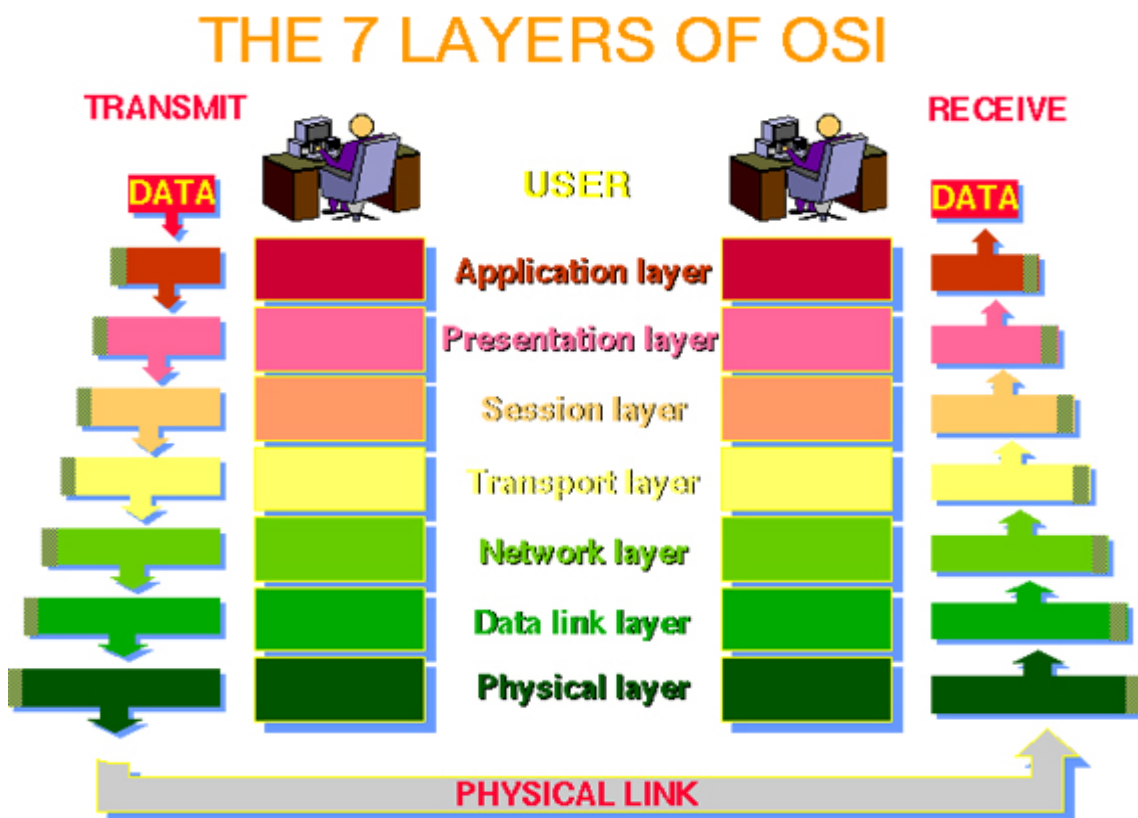


Figure 6 The Seven (7) Layers of Open System Interconnection (OSI) model created by the International Standards Organization (ISO).

The OSI model was specifically made for connecting open systems. These systems are designed to be open for communication with almost any other system. The model was made to break down each functional layer so that overall design complexity could be lessened. The model was constructed with several precepts in mind:

- a. Each layer performs a separate function;
- b. The model and its levels should be internationally portable; and
- c. The number of layers should be architecturally needed, but not unwieldy.

Each layer of the model has a distinct function and purpose as follows:

- a. *Application layer* - Provides a means for the user to access information on the network through an application. This layer is the main interface for the user to interact with the application and therefore the network. Examples include file transfer (FTP), DNS, the virtual terminal (Telnet), and electronic mail (SMTP);
- b. *Presentation layer* - Manages the presentation of the information in an ordered and meaningful manner. This layer's primary function is the syntax and semantics of the data transmission. It converts local host computer data representations into a standard network format for transmission on the network. On the receiving side, it changes the network format into the appropriate host computer's format so that data can be utilized independent of the host computer. ASCII and EBCDIC conversions, cryptography, and the like are handled here;
- c. *Session layer* - Coordinates dialogue/session/connection between devices over the network. This layer manages communications between connected sessions. Examples of this layer are token management (the session layer manages who has the token) and network time synchronization;
- d. *Transport layer* - Responsible for the reliable transmission of data and service specification between hosts. The major responsibility of this layer is data integrity--that data transmitted between hosts is reliable and timely. Upper layer datagrams are broken down into network-sized datagrams if needed and then implemented using the appropriate transmission control. The transport layer creates one or more than one network connection, depending on conditions. This layer also handles what type of connection will be created. Two major transport protocols are the TCP (Transmission Control Protocol) and the UDP (User Datagram Protocol);
- e. *Network layer* - Responsible for the routing of data (packets) to a system on the network; handles the addressing and delivery of data. This layer provides for congestion control, accounting information for the network, routing, addressing, and several other functions. IP (Internet Protocol) is a good example of a network layer interface;
- f. *Data link layer* - Provides for the reliable delivery of data across a physical network. This layer guarantees that the information has been delivered, but not that it has been routed or accepted. This layer deals with issues such as flow regulation, error detection and control, and frames. This layer has the important task of creating and managing what frames are sent out on the network. The network data frame, or packet, is made up of checksum, source address, destination address, and the data itself. The largest packet size that can be sent defines the maximum transmission unit (MTU); and
- g. *Physical layer* - Handles the bit-level electrical/light communication across the network channel. The major concern at this level is what physical access method to use. The physical layer deals with four very important characteristics of the network: mechanical, electrical, functional, and procedural. It also defines the hardware characteristics needed to transmit the data (voltage/current levels, signal strength, connector, and media). Basically, this layer ensures that a bit sent on one side of the network is received correctly on the other side.

Data travels from the application layer of the sender, down through the levels, across the nodes of the network service, and up through the levels of the receiver. Not all of the levels for all types of data are needed - certain transmissions might not be valid at a certain level of the model.

To keep track of the transmission, each layer "wraps" the preceding layer's data and header with its own header. A small chunk of data will be transmitted with multiple layers attached to it. On the receiving end, each layer strips off the header that corresponds to its respective level.

The OSI model should be used as a guide for how data is transmitted over the network. It is an abstract representation of the data pathway and should be treated as such.

Developers must realize there is more to programming than just code. There are two parts that address the important issue of application architecture using an N-tier approach (See [Figure 7](#) below). The first part is a brief introduction to the theoretical aspects, including the understanding of certain basic concepts. The second part shows how to create a flexible and reusable application for distribution to any number of client interfaces.

Technologies used consist of .NET (including C#, .NET Web Services, and symmetric encryption), Visual Basic, the Microsoft SOAP Toolkit, and basic interoperability (ability to communicate with each other) between Web Services in .NET and the Microsoft SOAP Toolkit. None of these discussions (unless otherwise indicated) specify anything to do with the physical location of each layer. They often are on separate physical machines, but can be isolated to a single machine. For starters, the terms "tier" and "layer" are used synonymously. "Tier" can be defined as "one of two or more rows, levels, or ranks arranged one above another". So from this, we get an adapted definition of the understanding of what N-tier means and how it relates to the application architecture: "Any number of levels arranged above another, each serving distinct and separate tasks." To gain a better understanding of what is meant, take a look at a typical N-tier model in [Figure 7](#) below.

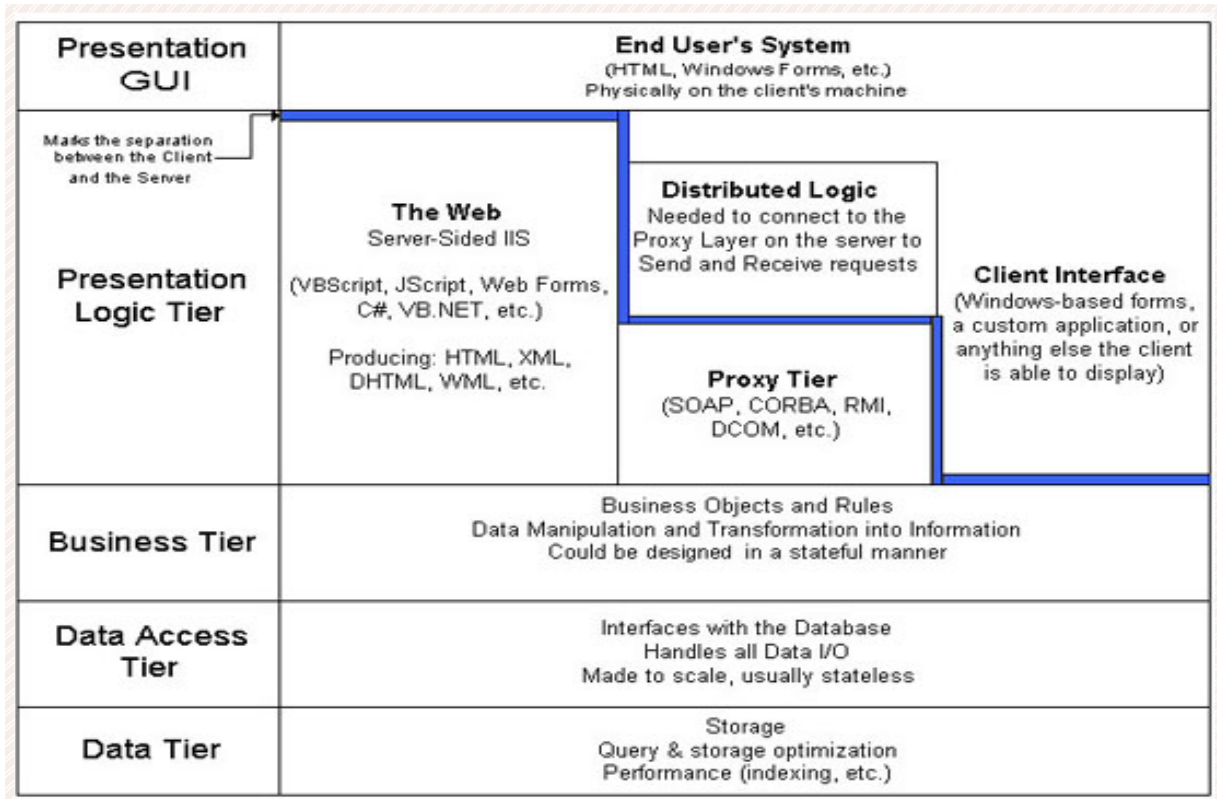


Figure 7 Typical N-Tier Application Model.

6.3.1 The Data Tier

Since this has been deemed the Age of Information, and since all information needs to be stored, the Data Tier described above is usually an essential part. Developing a system without a data tier is possible, but the data tier should exist for most applications. So what is this layer? Basically, it is the Database Management System (DBMS) - SQL Server, Access, Oracle, MySQL, plain text (or binary) files, etc. This tier can be as complex and comprehensive as high-end products such as SQL Server and Oracle, which do include things like query optimization, indexing, etc., all the way down to the simplistic plain text files (and the engine such as Objectivity[®] to read / search these files). Some more well-known formats of structured, plain text files include CSV, XML, etc. Note how this layer is only intended to deal with the storage and retrieval of information. It doesn't care about how it is planned on manipulating or delivering this data. This also should include the stored procedures: we should not place business logic in here, no matter how tempting.

6.3.2 The Data Access Tier

This layer is where someone will write some generic methods to interface with his data. For example, he will write a method for creating and opening a Connection object (internal), and another for creating and using a Command object, along with a stored procedure (with or without a return value), etc... It will also have some specific methods, such as "saveModel," so that when the Model object calls it with the appropriate data, it can persist it to the Data Tier. This Data Layer, obviously, contains no data business rules or data manipulation/transformation logic. It is merely a reusable interface to the database.

6.3.3 The Business Tier

This is basically where the brains of your application reside; it contains things like the business rules, data manipulation, etc... For example, if someone is creating a search engine and he/she wants to rate/weight each matching item based on some custom criteria (say a quality rating and number of times a keyword was found in the result), place this logic at this layer. This layer does not know anything about HTML, nor does it output it. It does not care about ADO or SQL, and it shouldn't have any code to access the database or the like. Those tasks are assigned to each corresponding layer above or below it.

A very basic understanding of Object-Oriented Programming (OOP) must be gained at this time. Object-oriented programming (OOP) is a programming language model organized around "objects" rather than "actions" and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data. The programming challenge was seen as how to write the logic, not how to define the data. Object-oriented programming takes the view that what someone really cares about are the objects he wants to manipulate rather than the logic required to manipulate them.

The first step in OOP is to identify all the objects someone wants to manipulate and how they relate to each other, an exercise often known as data modeling. Once someone has identified an object, he generalizes it as a class of objects and defines the kind of data it contains and any logic sequences that can manipulate it. Each distinct logic sequence is known as a method. A real instance of a class is called an "object" or, in some environments, an "instance of a class". The object or class instance is what someone runs in the computer. Its methods provide computer instructions and the class object characteristics provide relevant data. Someone communicates with objects - and they communicate with each other - with well-defined interfaces called *messages*.

The concepts and rules used in object-oriented programming provide these important benefits:

- a. The concept of a data class makes it possible to define subclasses of data objects that share some or all of the main class characteristics. Called inheritance, this property of OOP forces a more thorough data analysis, reduces development time, and ensures more accurate coding;
- b. Since a class defines only the data it needs to be concerned with, when an instance of that class (an object) is run, the code will not be able to accidentally access other program data. This characteristic of data hiding provides greater system security and avoids unintended data corruption;
- c. The definition of a class is reusable not only by the program for which it is initially created but also by other object-oriented programs (and, for this reason, can be more easily distributed for use in networks); and
- d. The concept of data classes allows a programmer to create any new data type that is not already defined in the language itself.

Smalltalk, C++ and Java, which are among the first object-oriented computer languages, are the most popular object-oriented languages today. The Java programming language is designed especially for use in distributed applications on corporate networks and the Internet. In advanced business applications, for example, in Banking and Finance, languages with advanced business-rules and expert-systems capabilities like PROLOG and LISP are extensively used. For example, Chase-Manhattan Bank uses a Prolog-based system to determine client-credit risk assessments. This type of object that encapsulates "intelligence" is called a software "agent".

Data modeling is the analysis of data objects that are used in a business or other context and the identification of the relationships among these data objects. Data modeling is a first step in doing object-oriented programming. As a result of data modeling, someone can then define the classes that provide the templates for program objects. A simple approach to creating a data model that allows someone to visualize the model is to draw a square (or any other symbol) to represent each individual data item that someone knows about (for example, a product or a product price) and then to express relationships between each of these data items with words such as "is part of" or "is used by" or "uses" and so forth. From such a total description, someone can create a set of classes and subclasses that define all the general relationships. These then become the templates for objects that, when executed as a program, handle the variables of new transactions and other activities in a way that effectively represents the real world. Several differing approaches or methodologies to data modeling and its notation have recently been combined into the Unified Modeling Language (UML), which is expected to become a standard modeling language.

6.3.4 The Presentation Logic Tier

Let's jump to the Presentation Logic Layer in [Figure 7](#) shown earlier. This layer consists of the standard ASP documents, Windows forms, etc... This is the layer that provides an interface for the end user into the application. That is, it works with the results/output of the Business Tier to handle the transformation into something usable and readable by the end user. It has come to the attention that most applications have been developed for the Web with this layer talking directly to the Data Access Layer and not even implementing the Business Tier. Sometimes the Business Layer is not kept separated from the other two layers. Some applications are not consistent with the separation of these layers, and it is important that they are kept separate. A lot of developers will simply throw some SQL in their ASP (using ADO), connect to their database, get the record set, and loop in their ASP to output the result. This is usually a very bad idea.

6.3.5 The Proxy Tier and the Distributed Logic

There's also that little, obscure Proxy Tier. "Proxy" by definition is "a person [object] authorized to act for another". This "object," in our context, is referring to any sort of code that is performing the actions for something else (the client). The key part of this definition is "act for another." The Proxy Layer is "acting" on behalf of the Distributed Logic layer (or end-user's requests) to provide access to the next tier, the Business Tier. Why would anyone ever need this? This facilitates the need for distributed computing. Basically it comes down in choosing some standard method of communication between these two entities. That is, "how can the client talk to the remote server?" This is where we find the need for the Simple Object Access Protocol (SOAP). SOAP is a very simple method for doing this. Without too many details, SOAP could be considered a standard (protocol) for accessing remote objects. It provides a way in which to have two machines "talking" or "communicating" with each other. (Common Object Request Broker Architecture [CORBA], Remote Method Invocation [RMI], Distributed Component Object Model [DCOM], SOAP, DIS, HLA, etc., all basically serve the same function).

6.3.6 The Client Interface

In this section of [Figure 7](#) it can be noticed that the end-user presentation (Windows forms, etc.) is connected directly to the Business Tier. A good example of this would be the applications over the Local Area Network (LAN). This is the typical, non-distributed, client-server application. Also notice that it extends over and on top of the Distributed Logic layer. This is intended to demonstrate how someone could use SOAP (or some other type of distributed-computing

messaging protocol) on the client to communicate with the server and have those requests be transformed into something readable and usable for the end user.

In reality, the Business Tier and Data Access Tiers are mostly combined tiers, allowing the Business Layer to talk directly to the Data Layer. The writing of the Data Access Tier, which is simply abstracting the Data Tier, may be an over-kill, and ADO can be considered as the Data Access Layer. It provides with the interface directly. It still keep all SQL in the Data Tier (stored procedures), but no business rules should be kept here.

Of course, the more tiers that is added, the more performance is affected. The path does affect performance. It is up to the application architect to know and understand this, and all other factors affecting the system, and be able to make a good decision on how to develop it at this level. This decision is usually pretty easily made, depending on the amount of work and documentation that was produced from the analysis phase.

It is now known how to do this logically. Let's explain the why. A good example is to look at the Presentation Logic Tier. Notice that many of its sections - the Web, the Proxy Tier, and the Client Interface - all sit directly on top of the Business Tier. It gains the advantage of not needing to redo any code from that Business Tier all the way to the database. Write it once, and plug into it from anywhere.

Now say someone is using SQL Server and he doesn't want to pay Microsoft's prices anymore, and he decides to pay Oracle's instead. So, with this approach he could easily port the Data Layer over to the new DBMS and touch up some of the code in the Data Access Layer to use the new system. This should be a very minimal touch-up. The whole point is to allow someone to plug each layer in and out (very modular) without too many hassles and without limiting the technology used at each tier.

7. The Modular M&S/SE Framework Overview

The proposed modular M&S/SE framework for developing and supporting distributed Collaborative Synthetic Environments (CSE) is designed to help DND in reducing costs and risks within specific programs as well as across Programs by leveraging from the harmonized, defined, and validated simulation concepts, models, tools, and utilities.

The general segmentation of a modular M&S/SE framework should be as follows:

- a Framework ([see section 9.1](#));
- b Simulation Runtime ([see section 9.2](#));
- c Software Development Environments (SDE) ([see section 9.3](#));
- d Client Applications ([see section 9.4](#));
- e Server Applications ([see section 9.5](#));
- f Distributed HLA Applications ([see section 9.6](#));
- g Management Applications ([see section 9.7](#));
- h Common Synthetic Environment (CSE) Infrastructure ([see section 9.8](#)); and
- i Dynamic Synthetic Environments/Computer Generated Forces (DSE/CGF) ([see section 9.9](#)).

7.1 Associated Services Requirements

In addition to defining a M&S/SE framework, it is strongly felt that there is also a requirement to define related Support Services to effectively support the user who wants to effectively use such a simulation, particularly in a CSE context. The general segmentation of M&S/SE Services required to be performed are the following:

- a Support Services ([see section 10.1](#));
- b Educational/Training Services ([see section 10.2](#)); and
- c Professional Services ([see section 10.3](#)).

The Services performed are an integral part of the modular M&S/SE framework delivery and are critical to the DND's intent to pursue efficient and effective distributed CSE, with support from Industry. Since Industry support is critical to DRDC's as well as DND's success, it was felt important to ensure that the Support Services that are part of the M&S/SE equation are appropriately documented in a holistic way.

7.2 DND Users Requirements

The DND users would require the following:

- a. Goods & Services that meets the DND modular M&S/SE framework;
- b. Complete set of After-market Support Services;

- c. Educational and Training Services;
- d. Professional Services;
- e. Featured Web Services;
- f. User/Developer Conferences;
- g. Communication Exchange Channels such as Newsletters, Best Practices, Examples, and FAQs;
- h. Advanced visibility to Clear Goods and Services Roadmaps;
- i. A Feature Request Process to capture DND existing and future Requirements and to influence how the Goods and Services will evolve;
- j. Discounting of the Goods and Services offered and yearly updated as the market prices change;
- k. Third-party access to the Goods or Services offered;
- l. Fostering a market driven strategy to enable the growth of M&S/SE in DND;
- m. Respect the Canadian standards on Intellectual Property management and rights;
- n. Publishing publicly an open business model policy to ensure access of the Goods and Services to Third-parties;
- o. Creating an Independent Software Vendor program in order to enable third-parties to develop applications based upon the modular conceptual M&S/SE framework; and
- p. Having the Vendor's IP based on PWGSC Standard Industry Terms and Conditions and Software License Agreement Terms and Conditions.

8. The Modular M&S/SE Framework Description

The modular M&S/SE framework relies on a network for communication between the various applications and legacy systems adapted to the framework. The modeling and simulation (M&S) software framework to support distributed Collaborative Synthetic Environments (CSE) depends on a layered, functionally separated approach to building dynamically reconfigurable applications. Each layer of the framework provides successive levels of specialization so that as new technology evolves, the implementation of the layer can be changed to accommodate new hardware or technology changes. For example, one of the lowest layers treats all operating systems as some abstract computer, called the "Generic OS Abstraction layer" so that the system is not tied to any one manufacturers operating system preference, but, can support and adapt to any and all manufacturers operating systems, now, and in the future. In addition, some of the layers are themselves complete infrastructures with their own domain specific architectures that open the possibilities for larger scale and wider deployment. An example is the Run Time Infrastructure (RTI) that directly provides the High Level Architecture (HLA) for real time communications of simulation coordination and control data.

HLA naturally has wider ranging influence in an M&S software architecture and its framework instantiation so that other layers such as the Simulation Runtime Services (SRS), which have common requirements captured and packaged in the Common Simulation Services (CSS) layer, provide controlled developments environments. The developer environment guards the policies of the framework defined by the Software Development Environment (SDE) layer through which developers can freely create the elements that will support entities that will run in the Common Synthetic Environment (CSE) infrastructure without worrying about "doing the wrong thing in the wrong place at the wrong time". The generalized CSE infrastructure is specialized into various modular forms such as Weather modules. These are not layers but are modular sub-systems that reside within or alongside other synthetic environment modules such as 911 Emergency Services and Communications. The core of the modular conceptual M&S/SE framework is its generality and its layers which may contain specialized infrastructures that are clearly, logically and functionally decoupled to support rapid application development for a system of systems approach. This approach is vital to network-centric simulations and software operations that support synthetic environments, distributed collaborative synthetic environments, and real-world operational interfaces, as suggested elsewhere (9). Therefore, since all applications are built upon the same solid basis framework, they are portable, interoperable, adaptable, evolvable, and reusable. Finally, the success of the modular conceptual M&S/SE framework is linked to the support provided by a management layer, support services, education and training services, and professional services.

The diagram below represents DND's intent for a modular conceptual M&S/SE framework and its associated services requirements:

MODULAR M&S/SE FRAMEWORK

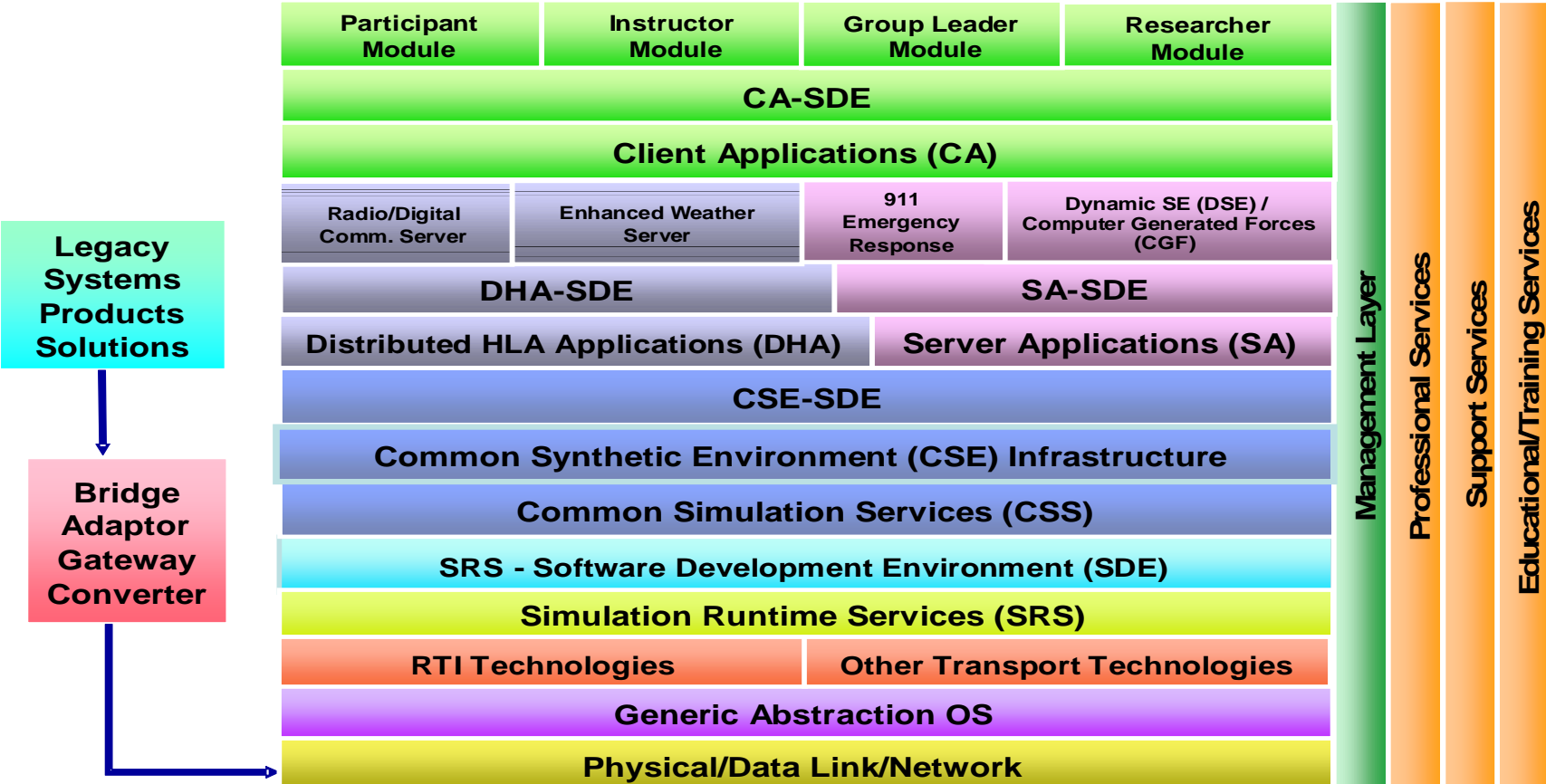


Diagram 1. Concept of a layered, de-coupled, modular M&S/SE open framework.

9. The Modular M&S/SE Framework Segmentation

The modular M&S/SE framework for developing and supporting distributed Collaborative Synthetic Environments (CSE) will be broken down into specific segments.

9.1 Framework

The modular M&S/SE framework shown above is segmented into a hierarchical layer structure that enables each layer to be replaced, modified, and/or upgraded without impacting other elements of the M&S/SE framework model. The M&S/SE framework is based on a Hierarchical and Relational object-oriented architecture that builds each component on top of each other in a maximally decoupled way to any other component.

The M&S/SE Framework is an object-oriented (OO) framework that should implement patterns for concurrent communication as well as a rich set of interfaces (facades) and other framework components that perform common intra-CPU communications (versus inter-CPU, between computers). The software should support tasks across a range of OS platforms and this means that the general framework includes, from a high level view, the following critical distinctions in the architecture, design and implementation:

- a. *Event Demultiplexing* - The job of gathering data from different application processes and packaging the data into one unit for transport over the network layer is called **multiplexing**. The job of delivering the data in a transport-layer to the correct application process is called **demultiplexing**. As per example, there could be three (3) forces in a simulation: a blue, red and green force. If blue and red fire separate missiles each onto the green force, then these different events need to be packaged as one unit and sent to the green force "target". When the event package arrives at the target, the software simulation engine needs to "unpack" the events i.e. to see who shot first and who is on target (demultiplexing step);
- b. *Event Handler Dispatching* - Each event is represented by an object that gives information about the event and identifies the event source. Event sources are typically entities or clients (users), but other kinds of objects can also be event sources e.g. other events. When events of different kind and number arrive from different sources, they must be collected, interpreted and managed. The event handler will collect and manage (i.e. handle) the events and each event type will usually require a handler dedicated to it. So, the dispatch of a handler deal with the events is very important in a large distributed application because this activity could be a bottle-neck if it is done incorrectly;
- c. *Signal Handling* - When a letter is typed on the keyboard, it generates a hardware interrupt to the CPU to process which letter that was hit. Signals are "software interrupts" and are needed so that an application process can handle events asynchronously. Signals are needed in order to support multiple "flow of control" in single-threaded processes as well as resource and logistics in multi-threaded situations;

- d. *Service Initialization* - Services are often made up of many different processes that each contributes some functionality to the realization or creation of the service. An object is required to recognize the type of service that a client requests and must support the management of the allocation of memory and other resources that the service needs as well as the needs of the individual objects that together provide the service. In a scaled distributed application, this process could eat up a lot of time or could cause bottle-necks, or worse, could create blocks so that the service itself uses up all the resources of the computer without ever providing the client the functionality requested. Initialization in this case is very important;
- e. *Interprocess Communication* - Interprocess communication (IPC) is designed to remove network layer communications and the latency and resource use that is related to this. IPC serve the needs of clusters of CPUs and local coordination and control through communication between processes that serve the needs of clients. A group of local application processes, perhaps even split up between 4 CPUs on a local mainframe need to be coordinated through the mechanism of IPC in order to effectively serve the client. It is common today to find many multi-processor machines to support the needs of M&S/SE;
- f. *Shared Memory Management* - For the purpose of reducing average memory access latency, shared memory is used by multi-processors or in multi-tasking operating systems for M&S/SE. Shared Memory uses a part of main memory distributed among clusters or tasks or CPUs or even processes as a cache or information exchange point to improve efficiency, conserve resources and support IPC. Shared memory situations are related to IPC as well as to remote clients (as in read-write locking of shared disk resources). LINDA and Tuple Spaces are paradigms for distributed parallel processing as well as models of shared data spaces that also require shared memory management. In this situation, multiple clients get (read) information from a virtual world but may also update it. These virtual worlds are themselves shared memory and data spaces and that is a resource so it needs to be managed; otherwise, broken or corrupted client processes could run crazily because of a software fault and use up all resources if management were not put in place;
- g. *Message Routing* - The greater the complexity of a system, the greater the chance for breaks in trust. When systems are highly distributed, the channels and paths from one network to another that preserve the trust must be protected and this is what message routing can do. Message routing arranges the paths of messages that originate in trusted networks to try to get to their endpoints by minimizing untrusted paths from the perspective of security. In other cases where security is not a problem, message routing permits the optimal arrangement for IPC or event handlers to get the events (messages) that they need with the least amount of distance (path length) and latency;
- h. *Dynamic Adaptation/Configuration of Distributed Services* - An example of dynamic configuration is in an environment where different clients (e.g. a PDA, a laptop and a mainframe) access the same service from a server, then the server must adapt the functionality and the way that the functionality is delivered to meet the requirements of the client. In the case of adaptation, when feature of the services are changed during execution and the software can deal with these changes (e.g. while purchasing a ticket using US dollars, the service configuration which includes hotel booking deals with French Francs, then these two different types are managed or adapted using a currency-conversion protocol so that the client using his MasterCard does not have to worry about it, and, only gets one form, in dollars for the total amount). In this

travel-agent scenario, the service adapts to the users needs (to shop in only one currency even though more than one is really used). This example is even more important in M&S/SE where adaptation can compensate for different operational devices as well as changes in the ongoing situation;

- i. *Concurrent Execution and Synchronization* - If two or more computers or processes are working on the same computation (e.g. calculating the flight path of an aircraft) then these two computers must work together harmoniously to solve the problem. This working together and delivering the result at the same time is called concurrency i.e. concurrent execution. Of course, two different computers that are calculating the flight controls for a plane need to do so in a synchronized way otherwise, if one computer move one aileron one way and the other is still calculating, then the plane will naturally fly out of control;
- j. *Dynamic Shared Secured Data Spaces and Parallelism Models* - In areas where clients share the same virtual or synthetic environment, they are in a shared data space. When security is important, then that shared space must also have multi-level security and so is called a secured data space. These spaces are defined because multiple computer systems must interoperate and calculate within these spaces (which exist from shared memory or LINDA or Tuple blackboards that are collections of shared memory from different computers but that looks like memory from one remote machine). The parallelism model is directly affected by the kind and type of shared secured data space model that is chosen (e.g. Tuple-Space machines versus support vector machines etc...); and
- k. *Advanced Information Services Components* - Advanced information services (AIS) include concepts such as meta-search, deep semantic search and categorization as well as ontology based or domain specific information processing. For example, chemical weapons are a different domain than nuclear weapons and so the way in which the information relevant to these different domains is analyzed is governed by the domain ontology and the entity taxonomy (chemical bomb versus neutron bomb versus hydrogen bomb versus atomic bomb). AIS also include dynamic concepts such as supply-chain logistics which can cost a lot of money if the transports and routes and logistics are poorly chosen. Getting food to an army is as important as getting the army to the battlefield. AIS help to solve these problems. AAR (After Action Review/Reporting) is another form of AIS.

9.1.1 The M&S/SE framework should have the following characteristics:

- a Physical/Data-link/Network/Transport as follows:
 - i. Support for Standards-based Networking Architectures and technologies used today within DND;
 - ii. Support for Local-Area Network (LAN) and Wide-Area Network (WAN) current technologies;
 - iii. Independent from the other layers to ensure support for evolving networking technologies such as Wireless Networking and others;
 - iv. Support TCP/IP and UDP/IP protocols for HLA/RTI solutions; and
 - v. Support for Gateway(s), Bridge(s), Adaptor(s) and/or Converter(s) based on the HLA IEEE-1516 standard (www.ieee.org), enabling legacy Applications/Solutions to connect to the M&S/SE framework.

- b Enabling Legacy application to be Interoperable as follows:
 - i. HLA application network software adapter using shared memory interface i.e. Native software;
 - ii. HLA application network software adapter offering an Application Programming Interface (API) i.e. Middleware;
 - iii. Protocol Adapter that bridges an existing legacy protocol to an HLA protocol;
 - iv. Support the common “DMSO RTI” technology (HLA-RTI-1.3 NG v6; from www.virtc.com) and must later support the IEEE-1516 standard;
 - v. Support the Real-time Platform Reference Federation Object Model as follows
 - 1) RPR-FOM v1.0; and
 - 2) RPR-FOM v2.0 (www.sisostds.org)
 - vi. Interoperability with any new compatible FOM definition i.e. FOM agile.
- c Open Architecture as follows:
 - i. Published Application Programming Interfaces (APIs);
 - ii. Modular (can be divided into components);
 - iii. Extensible i.e. adaptable (can add new components);
 - iv. Customizable i.e. evolvable and upgradeable (can replace or enhance existing components);
 - v. Third-party solutions can be integrated into the M&S/SE framework;
 - vi. Generic (can be used in different domains of expertise); and
 - vii. Network Architecture configurations capable as follows:
 - 1) Centralized;
 - 2) De-centralized;
 - 3) Distributed; and
 - 4) Mixed Architecture.
- d Open Programming Structure as follows:
 - i. Develop Applications using a software development environment that is integrated and cost effective [i.e. not requiring continual high expense after delivery or installation of a system];
 - ii. Develop following an M&S process with a suite of tools that help the developers to build and focus on reusable and interoperable models; and
 - iii. Access to a central, departmental or national repository of models that are reusable and interoperable.
- e Interoperable as follows:
 - i. Network as follows:
 - 1) Transport Protocol; and
 - 2) Exchanged Data (FOM).
 - ii. Application Programming Interface (API) as follows:
 - 1) A dynamic library (plug-in) system that enables the data definition of the API i.e. the Framework requires that the vendor actually have API capabilities;
 - 2) Allow the currently defined API to be expanded or replaced with new definition(s) i.e. the Framework requires that the vendor provides an API management system and development capabilities (extension of existing API is also critical);

- 3) At the Compiler/Linker level i.e. the Framework requires that the vendors provides .h and .lib files to reduce the DND and 3rd-party efforts to use the Development Licensed Software by the Developers internally or externally at the designated Licensee Location, to add, modify, further develop, extend and otherwise produce adaptations, enhancements, and improvements to the End User Licensed Software and other software applications or create derivative works; and
 - 4) At the Shared Memory level i.e. the Framework requires that the vendor provides method to share and exchange data at the system Shared memory level.
- f Cross-Platform support for Client, Server, Distributed HLA, and Management Applications should include:
- i. MS Windows 2000 Pro Sp2 or later;
 - ii. MS Windows XP Pro Sp1 or later;
 - iii. Linux Redhat 7.2 or later;
 - iv. Linux SuSE v8.0 or later;
 - v. IBM OS AIX 5L or later;
 - vi. SGI OS IRIX 6.5 or later;
 - vii. SUN OS Solaris version 8 OE or later; and
 - viii. HP-UX 11i or later.
- g Scalable as follows:
- i. Support for more than one concurrent Simulation or Scenario on one or many Server platform(s);
 - ii. Support variable fidelity level for models and simulations;
 - iii. System supports current DND processor platform technologies i.e. RISC, INTEL, and SPARC;
 - iv. Multiple processor environments are supported and leverage Symmetrical Multi-Processor (SMP) or Asymmetrical Multi-Processor (AMP) architecture; and
 - v. Simulation Domain Performance as follows:
 - 1) Allow for Multiple Services per Domain [It will be important for the Framework be able to support multiple diverse simulation services per domain whether these services are locally available within one centralized Server or fully distributed across the network infrastructure];
 - 2) Allow for Multiple Services Domain as follows:
 - a Requirement for multiple Simulation Services Domains within a networking infrastructure as to offer segregated services within their own domain containers but all accessible within the same network i.e. you may have a domain for Army, Navy or Air Force all on the Defence network that are completely fire walled from each other; and
 - b Requirement for redundancy or fault-tolerance or load balancing in situations where the need for persistent federations is required where the framework will have to be able to offer grouping of similar services that are presented as one to the user i.e. you could have 3 or 4 Weather Servers on a network broadcasting the same weather patterns to the whole simulation and be able to either load balance or offer redundancy if one of them goes out of service.

- 3) Allow for Multiple Simulations per domain [Within a single Simulation Domain the Framework must be able to offer multiple simulation rings or segregated simulations as to avoid multiplication of hardware and the management of too many domains. In addition this capability is directly related to the requirement for a Server or set of Servers to operate multiple simulations in parallel];
- 4) Allow for Multiple Distributed HLA Applications per domain [The concept is for HLA applications could act a federate and could be advertised as part of the simulation domain so that a user again would not have to be concerned of where or how the simulation services are provided. So the query here is to better understand from the vendor if he can integrate HLA applications into his domain management and to provide to users a single view of all resources available during a simulation within a domain.];
- 5) Allow for Multiple entities per simulation [Here we are concerned by how many entities per simulations within a single domain a vendor solution will be able to support. The quantity of entities per simulation may directly affect the number of simulations and/or domains required in the architecture.];
- 6) Allow for Multiple computers/systems supported in a Simulation domain [Here we are concerned with understanding as part of a vendor solution how many computing systems can be part of a single domain. This is important in a single domain concept, as there could be a fair number of systems (providing Live, Constructive and Virtual Simulation Services) distributed across Canada over a Simulation Network]; and
- 7) Support HLA Data Management (regions).

h Generic Usage as follows:

- i. Collaborative Capability Management;
- ii. Concept Development and Experimentation (CDE);
- iii. Doctrine Analysis;
- iv. Requirements Analysis;
- v. Functional Analysis;
- vi. Operational Analysis (OA);
- vii. Full Spectrum Life-cycle requirements Analysis;
- viii. Model Development as follows:
 - 1) Physical;
 - 2) Process;
 - 3) Human Performance/Behaviour Representation;
 - 4) Maintenance Logistics/Cost; and
 - 5) Human Machine Interface Requirement.
- ix. Scientific Research, Development & Engineering;
- x. Test and Evaluation (T&E);
- xi. Material Acquisition and Support (MA&S);
- xii. Training and Rehearsal; and
- xiii. Verification, Validation & Accreditation (VV&A).

i Extensible as follows:

- i. Conceptual types providing functionality that is not necessarily part of all programming languages i.e. decoupling of functionality and implementation which means that there is support for the concept of an Ontology and/or Taxonomy of concepts and their types;
 - ii. Non-aggregate types providing support for the basic types of the programming languages as well as for new ones; and
 - iii. Aggregate types providing support for the complex data types that are specific to the applications.
- j Human-oriented design (user-friendly) i.e. GUI-Oriented and making maximal use of intelligent user support aids such as “Wizards”;
- k Licensing mechanism and proper licensing methods and system should be in place as follows:
 - i. Individual (seat);
 - ii. Floating (server);
 - iii. Instances;
 - iv. Multi-Level;
 - v. Departmental;
 - vi. Site;
 - vii. Governmental; and
 - viii. Time Expiring.
- l Upgradeability as follows:
 - i. Each layer is upgradeable independently from each other as to ensure investment protection for DND;
 - ii. Legacy Applications can be upgraded without impacting their Interoperability with the M&S/SE framework i.e. the legacy application and *not* the SE framework that is adapted
 - iii. Client, Server, Distributed HLA, and Management Applications are upgradeable without impacting each other; and
 - iv. Third-Party applications are upgradeable without impacting the vendor’s Applications.
- m Methods for Software Goods Distribution as follows:
 - i. Major Release;
 - ii. Minor Releases;
 - iii. Service Packs;
 - iv. Upgrades/Updates;
 - v. Emergency Software Releases;
 - vi. Demonstrator or Evaluation;
 - vii. Site Master CD;
 - viii. Web Download; and
 - ix. FTP Server Download.
- n Provision to implement Security and Access control capabilities;
 - i. Enabled to adapt to Security requirements of DND as they evolve.
- o Network Centric as follows:

- i. Support HLA-RTI protocol for simulation data transport;
 - ii. Support FOM agility without affecting application models;
 - iii. Has Portable data types to abstract the modelers of cross platform issues; and
 - iv. Capacity to support legacy protocol such as Distributed Interactive Simulation (DIS) and other to ensure reusability and investment protection, through the use of Adaptors, Gateways, Bridges and/or Converters.

- p Multiple Language Support as follows:
 - i. Support for both official languages - English and French;
 - ii. Two Bytes Unicode character is supported; and
 - iii. Multiple units system is supported.

- q Professional Documentation as follows:
 - i. Produced professionally by a technical writer and includes the following subject matter:
 - 1) General Applications:
 - a. Introduction;
 - b. Frequently Asked Questions (FAQ's);
 - c. Getting Started;
 - d. Acronyms and Terminology;
 - e. Quick References;
 - f. Technical Specifications;
 - g. Installation Guide;
 - h. Configuration Guide;
 - i. Management Guide;
 - j. User's Guide;
 - k. Release Notes;
 - l. Known Problems;
 - m. Online Help;
 - n. Release Notes; and
 - o. White Papers.
 - 2) Development Environments as follows:
 - a. Programmer's Guide;
 - b. Migration/Conversion Document;
 - c. On-line hyperlinked Help class hierarchy;
 - d. UML diagrams;
 - e. Reference Manual; and
 - f. White Papers.

9.2 Simulation Runtime

The Simulation Runtime should provide all basic Simulation Runtime services required to operate Generic Simulations. The architecture of the simulation runtime should be divided into two (2) layers as follows:

- a. A layer which includes the services necessary to support an expandable and configurable system; and
- b. A layer which defines how the application components perform their processing and how they interact with one another.

9.2.1 The first layer would require the following:

- a. Professional Documentation as follows:
 - i. Introduction;
 - ii. Getting Started;
 - iii. Frequently Asked Questions (FAQ's);
 - iv. Acronyms and Terminology;
 - v. Quick References;
 - vi. System Requirements;
 - vii. Network Requirements;
 - viii. Infrastructure Requirements;
 - ix. Generic OS Requirements;
 - x. RTI Requirements;
 - xi. Product Installation;
 - xii. Configuration;
 - xiii. Operation;
 - xiv. Management;
 - xv. Administration;
 - xvi. Debugging; and
 - xvii. Known Problems.
- b. Basic type definitions, viewable in a user accessible Ontology, to make the system types portable with additions as follows:
 - i. Define portable equivalent to the system basic types i.e. C++ requires additional components to interoperate and the vendor will have to ensure that these additional plug-ins and components are available as part of the Simulation Runtime;
 - ii. Define new types i.e. Basic types are too generic and the framework will require that users can create custom type as to add the appropriate additional functionality required;
 - iii. Add commonly used mathematical data types; and
 - iv. Support Internationalization as follows:
 - 1) Names such as Unicode Character String; and
 - 2) Representation and Data Units.
- c. Repository management services for storing application information as follows:
 - i. Local Repository as follows:
 - 1) At the object or component level for object-specific runtime information (Log);
 - 2) Capable to select which variables to log at runtime; and
 - 3) At framework level for global information.
 - ii. Central Repository (Data Warehouse);
 - iii. SQL DBMS format and engine;

- iv. Commercial DBMS based on a modern Object-Oriented DBMS in line with the object-oriented paradigms of the framework;
- v. Shared/Distributed Repository;
- vi. Import/Export using XML standard schemas for single objects parameters, objects composition, scenarios, and other object management data; and
- vii. Provide basic manipulations for the following:
 - 1) Data entry;
 - 2) Data import;
 - 3) Data export;
 - 4) Backup utilities;
 - 5) Transaction mechanism for data integrity;
 - 6) Data access logging and control for security;
 - 7) Data change notification to automatically refresh displays; and
 - 8) Data versioning to keep track of changes.
- d. Instance management services for handling ready to re-use instances should be as follows:
 - i. Provide services for creating, modifying and deleting ready to re-use instances;
 - ii. Provide a mechanism for selecting instances based on selection criteria; and
 - iii. Organize the instances into a hierarchical view-based structure such as folder-based structure.
- e. Data Type management services for handling application defined types should be as follows:
 - i. Provide services to manipulate the type definitions including attributes and methods;
 - ii. Provide services to access the data within the attributes of the application instances for the purpose of debugging and validation;
 - iii. Provide services to create instances of the application types without having access to the sources of the application; and
 - iv. Provide services for the controlling the types usage for the purpose of licensing i.e. Licensing Enforcement. The Framework will require Instance control i.e. control of the number of applications and/or module that can concurrently run at the same time, based on the licensing grant and method.
- f. Composition management services for creating complex objects should be as follows:
 - i. Define the mechanism needed to compose a complex entity by associating small elements. The small elements are organized in a relational structure to represent real world entities and their parts;
 - ii. Define a mechanism to describe what constitutes a valid relational structure for a particular kind of entity. This is to ensure that only valid entities are created;
 - iii. Provide an initialization mechanism for the entities that allows the use of external data in XML format and linked to the repository. The initialization data may come from a certified source while the implementation may come from another; and

- iv. Provide a conversion mechanism to convert the certified initialization data into an acceptable format for the implementation.
- g. User management services for identifying the various users of the framework should be as follows:
 - i. Provide services to maintain a list of the valid user of the application;
 - ii. Provide services to verify the legitimacy of a user accessing the application;
 - iii. Provide services to store the user's preferences (Internationalization); and
 - iv. Provide entry points for addition of user level monitoring i.e. The framework would have to provide for the creation of additional user level monitoring Features and Functionalities in order to better monitor usage for auditing and performance purposes.
- h. Security management services for controlling access to information should be as follows:
 - i. Provide services to assign privileges to users i.e. User Definition;
 - ii. Provide services for grouping users for the purpose of simplifying privilege assignment;
 - iii. Provide services for controlling the access to information i.e. Level Definition. The Framework requires that the vendor solution provides for a dynamic loading of plug-ins. However, each instance will have to be recognized prior to execution.; and
 - iv. Provide services for validating the privileges of a user for the purpose of controlling the access to information i.e. Enforcement (Access).
- i. Plug-in management services to allow implementation extensions should be as follows:
 - i. Provide services to identify and access the application libraries;
 - ii. Provide services to limit the access the application libraries only when they are required;
 - iii. Provide services to access a new application library after the application has started executing; and
 - iv. Provide services for controlling library usage for the purpose of licensing.
- j. Internal Server management services for adding global or shared functionalities should be as follows:
 - i. Provide services to control access to the API (application programming interface) associated with the shared application services; and
 - ii. Provide services to select an appropriate implementation when multiple implementations of the shared services are available. As per example, if there is a method written in FORTRAN, in JAVA and in C that all implement the same function, sometimes the FORTRAN may be more appropriate when the application has access to a high speed array number processor, and on other occasions, the JAVA version may be selected when reduced security and lower criticality are more important. It all depends on priority of services (e.g. also security prioritization based on MLS). SOAP protocol, .NET and other such formats are designed to provide the same solution to the issue of different

implementations to meet other requirements other than functionality (e.g. if CONTROL of performance is a requirement).

- k. Adapter management services for adding functionalities to local objects should be as follows:
 - i. Provide services to control access to the application programming interface (API) associated with the extended object services; and
 - ii. Provide services to select an appropriate implementation when multiple implementations of the extended object services are available. The CORBA specification as well as other distributed specifications [e.g. XML based-service invocation such as Simple Object Access Protocol (SOAP) and Web Services Flow Definition Language (WSFDL)] recognize the different implementations of the same service and also provide the way to provide configurations for different hardware (e.g. Palm Pilot versus desktop computer). GIIOP, IIOp, BOA and POA are all mechanisms in CORBA to help inter-object-protocols (IIOp) deal with the issue of multiple services with multiple hardware and with multiple performances per functionality requirements (QOS) criteria. All Adapter Management Services are brokers.

9.2.2 The second layer would require the following:

- a. Professional Documentation as follows:
 - i. Introduction;
 - ii. Getting Started;
 - iii. Frequently Asked Questions (FAQ's);
 - iv. Acronyms and Terminology;
 - v. Quick References;
 - vi. System Requirements;
 - vii. Network Requirements;
 - viii. Infrastructure Requirements;
 - ix. Generic OS Requirements;
 - x. RTI Requirements;
 - xi. Product Installation;
 - xii. Configuration;
 - xiii. Operation;
 - xiv. Management;
 - xv. Administration;
 - xvi. Debugging; and
 - xvii. Known Problems.
- b. Time management services for controlling how simulation time advances should be as follows:
 - i. Provide a reference clock such as Atomic or GIS clocks;
 - ii. Provide logical clocks such as Lamport clocks;
 - iii. Support "as fast as possible" for batch runs;
 - iv. Allow the simulation time to advance in real-time (at the same rate as the system clock). It may also allow the simulation time to advance at a rate faster

- or slower but proportional to real-time. The advancement of time may be triggered by the system clock or by an external system event;
- v. Allow the simulation time to advance in predictable time thus insuring that very short execution rate are possible, and that delays are minimal between the requested start time and the actual start time;
 - vi. Allow the simulation time to advance in soft real-time (at a fixed rate). Distributed application is to be partially synchronized to ensure that the simulation time in the various processes do not continuously drift apart;
 - vii. Allow the simulation time to advance in synchronized-time thus ensuring that the simulation times in the various processes of a distributed application are always identical;
 - viii. Provide services to freeze or suspend the execution of the simulation;
 - ix. Allow synchronization points to be used by the application. They are used to ensure that all processes in a distributed application have completed some processing before the simulation time advances and each of the processes are allowed to continue with their next processing;
 - x. Provide the implement of HLA Time Management; and
 - xi. Provide a recovery mechanism to handle case whereas the processing within an iteration and within a process takes longer than the allowed time. This is known as the overrun condition.
- c. Iteration scheduling services for controlling continuous processing should be as follows:
- i. Provide services to allow each object in the application to specify its own execution rate. The rate may be limited to be relative to the time management rate but this rate is not to be limited except that the computing load may be affected;
 - ii. Provide services to allow each object in the application to specify a priority in relation to the other objects;
 - iii. Allow the processing within a single process to be divided among the available processors of a computer. This is normally achieved through the use of processing threads combined with a locking mechanism to control access to shared data;
 - iv. Allow the processing to be distributed among several computers without HLA; and
 - v. Provide services to specify if the processing occurs even if the simulation is frozen.
- d. Interaction scheduling services for controlling punctual processing should be as follows:
- i. Provide services to indicate when the interactions are to be delivered (immediately or in the future); and
 - ii. Provide services to specify if the interactions are to be as follows:
 - 1) Broadcasted (all targets);
 - 2) Multicast (multiple targets); or
 - 3) Targeted (single target).

- e. Interest management services for data publishing and subscribing using “publish-subscribe” Broker system should be as follows:
 - i. Provide services to access all the instances of a specified type, including the derived types, that are published;
 - ii. Provide services to filter the list of published instances. Filtering may be based on value range on the attributes of the instance; and
 - iii. Provide services to allow notification to occur when the list of published instances changes.
- f. Ownership management services for exchanging control of information should be as follows:
 - i. Provide services to take ownership of the attributes of a local object (both objects are in the same process) i.e. Transfer within a process;
 - ii. Provide services to take ownership of the attributes of a remote object (the two objects are in different processes) i.e. Transfer across a process; and
 - iii. Provide services to control and manage the broker security and authentication levels.
- g. Network management services for data exchange over a network should be as follows:
 - i. Support multiple protocols;
 - ii. Provide services to initiate and terminate transfer of information over the network;
 - iii. Provide services to control multiple simultaneous connections over various networks;
 - iv. Provide various services to support the other components of the system that require assistance in handling distributed application; and
 - v. Provide services for monitoring and introspection of network services.
- h. FOM agility services to facilitate interoperability at the data level should be as follows:
 - i. Provide services to select what kind of data is to be transferred over the network; and
 - ii. Provide services to convert the application data into the selected kind of data.
- i. Scenario management services for handling application definitions should be as follows:
 - i. Provide services to create and delete scenarios;
 - ii. Provide services to open and close existing scenarios;
 - iii. Provide services to manage batch runs;
 - iv. Provide services to modify a scenario by adding and removing processes; and
 - v. Provide services to modify a scenario by adding and removing objects within the processes of the scenario.
- j. Federate management services for controlling the application execution should be as follows:
 - i. Provide services to load and unload a federate from a scenario definition;
 - ii. Provide services to select on which computers the processes will be executing;

- iii. Provide services to control the execution of a federate. This includes resuming and pausing an exercise;
 - iv. Provide services to manipulate the content of a federate by adding and removing processes;
 - v. Provide services to manipulate the content of a federate by adding and removing object within a process;
 - vi. Provide services to move an object from one process to another to balance the load between the available computers;
 - vii. Provide services to save the state of a federate into a new scenario definition;
 - viii. Provide services to record and playback the execution of a federate; and
 - ix. Provide services to move objects between processes.
- k. Debugging services for verifying and validating the application should be as follows:
- i. Provide services to query the content of the local process and of the remote processes;
 - ii. Provide services to access and modify the attributes of the application objects;
 - iii. Provide services to collect and drives over time, the attributes of the application objects;
 - iv. Provides services to export the collected data in XML format; and
 - v. Provide services to manage the information collected from process verification and validation.
- l. Device management services for controlling access to system devices should be as follows:
- i. Provide services to determine the available devices;
 - ii. Provide services to reserve and release devices thus ensuring that the devices are used in an orderly fashioned;
 - iii. Provide services to access the most commonly used device i.e. joystick, sound card, and trackballs; and
 - iv. Provide access to specialized buses or devices.
- m. Resource management services for controlling resource usage should be as follows:
- i. Provide services to determine the available resources;
 - ii. Provide services to determine the state of the resources i.e. resource monitoring; and
 - iii. Provide services to reserve and release Resources such as computers and processors.

9.3 Software Development Environments (SDE)

Within the M&S/SE framework, there should be Software Development Environments (SDE) that allows developers to create specialized applications or add/extend and/or replace a vendor's applications at different layers level of the M&S/SE framework as follows:

- a. Simulation Runtime Services (SRS) SDE (SRS-SDE);
- b. Common Synthetic Environment (CSE) Infrastructure SDE (CSE-SDE);
- c. Client Applications (CA) SDE (CA-SDE);
- d. Server Applications (SA) SDE (SA-SDE), and
- e. Distributed HLA Applications (DHA) SDE (DHA-SDE).

9.3.1 Support for M&S/SE Concepts and Attributes

The aforementioned SDEs should support the M&S/SE framework Concepts and Attributes included in section [9.1.1](#) of this document as follows.

- a. Physical/Data Link/Network/Transport;
- b. Enabling Legacy Applications;
- c. Open Architecture;
- d. Open Programming Structure;
- e. Interoperable;
- f. Generic Usage;
- g. Extensible;
- h. Human Oriented Design;
- i. Licensing Mechanism;
- j. Upgradeability;
- k. Software Goods Distribution Methods;
- l. Provision to implement Security and Access Control;
- m. Network Centric;
- n. Multiple Language Support; and
- o. Exceptions to the M&S/SE framework should be as follows:
 - i. Cross-Platform - The SDE support Windows 2000 Professional Server and Workstation; and
 - ii. Scalable - The SDE support Multi-Processor environments as per the M&S/SE framework.

9.3.2 All Software Development Environments (SDE) should include the following:

- a. Professional Documentation for the following:
 - i. Programmer's guide describing how to develop applications that are based on the M&S/SE framework;
 - ii. Migration document describing the changes between each release of the M&S/SE framework and also describes how to apply the changes;
 - iii. Reference manual that describes the services available from the M&S/SE framework;
 - iv. Provide a series of white papers describing in more details how specific features can be used to implement advanced concepts;
 - v. Documentation requirements are as follows:
 - 1) Technical Specification;
 - 2) Release Notes;
 - 3) Programmer's Online Help;
 - 4) Programmer's Guide;

- 5) Debugging;
 - 6) Known Problems;
 - 7) Programmer's Training Course (with training); and
 - 8) Migration/Conversion Document (when applicable).
- vi. Supporting Documentation for Developers in addition to the Development Documentation should be as follows:
- 1) Introduction;
 - 2) Quick References;
 - 3) System Requirements;
 - 4) Infrastructure Requirements;
 - 5) Generic OS Requirements;
 - 6) RTI Requirements;
 - 7) User Interface Requirements;
 - 8) Product Installation;
 - 9) Getting Started;
 - 10) Configuration;
 - 11) Operation;
 - 12) UML diagrams;
 - 13) Management; and
 - 14) Administration.
- b. Development Language for the following:
- i. Allow components to be developed using the Java programming language;
 - ii. Allow component to be developed using the C++ programming language; and
 - iii. Allow foreign language interface support such as FORTRAN, Prolog, Lisp, and Ada.
- c. Development System for the following:
- i. Allow components to be developed under the Microsoft Visual Studio development system on a Microsoft Windows operating system;
 - ii. Allow components to be developed under the GNU development system on a UNIX based operating system;
 - iii. Allow components to be developed under the Borland development system on a Microsoft Windows operating system; and
 - iv. Provide a set of include files and libraries needed to compile and link the application code into plug-ins under all development systems. These plug-ins are used by the simulation framework runtime to create applications.
- d. Modeling System for the following:
- i. Allow components to be developed under the Rational Rose® modeling system;
 - ii. Allow components to be developed under the Together® ControlCenter™ modeling system; and
 - iii. Allow components to be developed under the MATLAB® and Simulink® modeling system.
- e. Source Code Generators for the following:

- i. Provide wizard-type tools for creating the skeleton of the plug-ins that are used by the application;
 - ii. Provide wizard-type tools for creating the skeleton of the classes that are used by the application;
 - iii. Provide wizard-type tools for adding attributes to the classes that are used by the application;
 - iv. Provide a graphical tool for creating a FOM specification within the context of the FOM agility component. It allows for specifying the mapping between classes and between attributes. It is possible to specify standard conversion method when mapping attributes; and
 - v. Wizards integrated with the development systems or with the Modeling system listed above in sub-paragraph d.
- f. Debugging Tools for the following:
 - i. Provide a debugging tool to allow the application to be debugged and validated;
 - ii. Debugging tool integrated with the client shell as an Add-on thus providing the same look and feel as the other tools;
 - iii. Use the debugging services from the simulation framework runtime to access the application information;
 - iv. Allow the application information to be monitored and modified;
 - v. Allow the application information to be collected and driven. To validate an application component, its inputs are driven with known values while its output values are collected for analysis;
 - vi. Allow the collected application information to be stored for later review;
 - vii. Allow the collected application information to be displayed in graph form. It must be possible to overlay the expected values on the same graph as the collected values to visually inspect their correlation;
 - viii. Allow the application scheduling to be monitored and controlled. This includes the interaction and the interactions scheduling;
 - ix. Provide information processing and filtering tools to manage the collected data for visualization or other methods of analysis;
 - x. Allow to connect with or develop new analysis tools; and
 - xi. Allow the results to be exported in a standard XML format.
- g. Coding examples as follows:
 - i. Provide a set of entry-level example showing the use of the most commonly used features of the simulation framework runtime; and
 - ii. Provide a set of advanced example showing the use of the less used features of the simulation framework runtime.
- h. Test/Quality assurance (QA)/Performance as follows:
 - i. Allow Commercial-off-the-shelf (COTS) tools that are supported by the development system to be used for testing/QA and performance analysis. However, such tools may have an impact on the execution of the simulation. Such as Intel's VTUNE and Parasoft's complete tools which are the most common used.
- i. Configuration Management as follows:

- i. Allow COTS tools that are supported by the development system to be used for managing the development of software.
- j. Access to Third- Party Libraries/Frameworks as follows:
 - i. Allow 3rd Party libraries that were developed for the simulation run-time to be used. Such third party libraries must be developed based on the same set of requirements as the simulation run-time. The Third-party libraries may have an impact on the performance of the simulation; and
 - ii. Allow Third-Party libraries that were not developed for the simulation run-time to be used. However, the use of such libraries must not prevent the normal execution of the simulation.
- k. Management Tools enabling the support of the Management Applications Requirements are detailed in section [9.7.2](#) of this document.

9.4 Client Applications

Client applications are meant as a set of tools and GUIs that should perform certain tasks and/or provide a certain set of Services within simulation as follows:

- a. Connect to the Simulation Domain and/or Server Application(s);
- b. Present Information provided by the Simulation Domain and/or Server Application(s);
- c. Provide Input and Control Capabilities of the Environment;
- d. Manage Information Flow and Dissemination; and
- e. Control Aspects of the Simulation, Entities or Applications.

9.4.1 DND User Community

These Client Applications should support the Client/Server concepts in order to isolate them from Server and Distributed HLA Applications and to enable a wide-ranging set of Client applications to support the DND user community, which is as follows:

- a Decision-Makers;
- b Program Leaders;
- c Project Leaders;
- d Project Directors;
- e Project Managers;
- f Group Leaders;
- g Systems Engineers;
- h Researcher;
- i Scientists;
- j Analysts;
- k Instructors;
- l Simulation Managers;

- m Scenario Managers;
- n Exercise Managers;
- o IT/Network Managers;
- p Subject Matter Experts (SME);
- q Hardware/Software Developers;
- r Technical Support Managers;
- s Technicians; and
- t Participants.

9.4.2 Support for M&S/SE Concepts and Attributes

The aforementioned Client Applications support the Concepts and Attributes of the M&S/SE framework detailed in section [9.1.1](#) of this document as follows:

- a. Physical/Data Link/ Network/Transport;
- b. Enabling Legacy Applications;
- c. Open Architecture;
- d. Open Programming Structure;
- e. Interoperable;
- f. Cross-Platform;
- g. Scalable;
- h. Generic Usage;
- i. Extensible;
- j. Human Oriented Design;
- k. Licensing Mechanism;
- l. Upgradeability;
- m. Software Good Distribution Methods;
- n. Provision to implement Security and Access Control;
- o. Network Centric; and
- p. Multiple Language Support.

9.4.3 The Client Applications should also include the following:

- a. Documentation should have the following:
 - i. Written by a Professional Technical Writer;
 - ii. User's guide containing description how to use the clients;
 - iii. Online help containing clarification on how to use the clients;
 - iv. White papers containing descriptions on how to implement advanced concepts; and
 - v. The Professional Documentation covers the following subjects:
 - 1) Introduction;
 - 2) Acronyms and Terminology;
 - 3) Quick References;
 - 4) System Requirements;
 - 5) Infrastructure Requirements;
 - 6) User Interface Requirements;
 - 7) Generic OS Requirements;

- 8) RTI Requirements;
 - 9) Product Installation;
 - 10) Getting Started;
 - 11) Configuration;
 - 12) Operation;
 - 13) Management;
 - 14) Administration;
 - 15) Problem Solving; and
 - 16) Known Problems.
- b. Support for Online Help Files;
 - c. Generic Instance Editor Add-on for the following:
 - i. Manipulate the ready to re-use instances;
 - ii. Allow the ready to re-use instances to be created, modified and deleted; and
 - iii. Allow the viewing and editing of the instance internal organization. An instance is composed of a hierarchy of small objects.
 - d. Generic Scenario Editor Add-on for the following:
 - i. Manipulate the scenario definitions; and
 - ii. Allow the scenario definitions to be created, modified and deleted.
 - e. Generic Exercise Controller Add-on for the following:
 - i. View the content and to control the execution of the exercise; and
 - ii. Allow the processes and the objects in the processes to be manipulated. This includes adding and removing processes and objects.
 - f. Generic Ontology Editor Add-on for the following:
 - i. View and edit meta-data definitions [Meta-data definition is a new an emerging field that is a proper sub-field of ontology creation, design and editing. Meta-data is data about the data (for example: The "vehicle" could classify "car" and "truck". That is the ontology but if we were to look at the Meta-Data, then we could say the "car" term has the attribute meta-data of "model" and then "model" could refer to "Ford, BMW, etc..." So as it can be seen, the difference is subtle but fundamental to modeling databases today. In fact, to understand more, you just have to search it on Google];
 - ii. View the data definitions for entities, environments and other critical conceptual data [Data definitions are a part of the data definition language (DDL) which is a part of any databasing system. A viewer is something that is used to see these things. Critical conceptual data would be any complex piece of information that defines something that resides or controls (directly or indirectly) the simulation world.]; and
 - iii. Allow the processes and the objects in the processes to be controlled by parameters. This includes adding and removing control parameter information relative to the objects for modifying runtime behaviour. [Parametric objects are a characteristic of Ontology driven, Model-Driven Architectures (MDA). MDA is the preferred way to build modern object systems for simulations. The parameter could be "model" number and the object could be "car". So

dynamically at runtime, someone could define the vehicle object to be a BMW car. Again, parametric objects are a field that is well understood in most advanced OO systems and used (see Design Patterns and other notes on internet about parametric objects). When a parametric object contains another object, then it results in having a meta-object i.e. the parameters are like the "meta-data"]].

- g. Entity Editor Add-on for the following:
 - i. Manipulate entity definitions. The entity definitions are a specialized sub-set of the ready to re-use instances;
 - ii. Allow the creation and deletion of entity definitions;
 - iii. Allow the viewing and editing of the entity internal organization. An entity is composed of a hierarchy of small objects;
 - iv. Use a 3-dimensional (3D) representation of the entity to manipulate the organization of the entity. This allows a user, among other things, to drag and drop equipments onto the appropriate location on the entity or to select an existing component and delete it;
 - v. Allow the viewing and editing of the characteristics of the entity; and
 - vi. Use the composition management component from the simulation runtime to validate and manipulate the entity [in this context, validate means to make sure that initialization, definitions and parameters are correct for a specific HLA accessible entity].
- h. Behaviour Editor Add-on for the following:
 - i. Manipulate the behaviour descriptions that can be assigned to entities; and
 - ii. Allow the behaviour descriptions to be created, modified and deleted.
- i. Entity Controller Add-on for the following:
 - i. Manipulate the entities during the execution of the application;
 - ii. Allow the viewing and editing of the characteristics of the entity;
 - iii. Allow the entity to be controlled. The type of controls depends on the type of the entities;
 - iv. Allow the behaviour assigned to an entity to be monitored;
 - v. Allow the behaviour to be manually controlled; and
 - vi. Allow manipulation layers to be created that are specific to the type of the entities. Each type of entity then have specialized manipulations available for them.
- j. Geographical Situation display (2D) Add-on for the following:
 - i. Manipulate the scenarios and the exercises that have an associated terrain. The terrain becomes the bases for a more intuitive user interface;
 - ii. Allow entities to added or removed from the scenarios or the exercises;
 - iii. Allow the symbology to be selectable by the user; and
 - iv. Allow information layers to be created. They can then be activated or de-activated by the user when needed.
- k. Stealth View Display (3D) Add-on for the following:
 - i. View the exercises that have an associated terrain;

- ii. Allow the symbology to be selectable by the user; and
 - iii. Allow information layers to be created. They can then be activated or de-activated by the user when needed.
- l. Debrief Station Add-on to review the details of a simulation execution;
 - m. After Action Review Add-on to review the result of a simulation execution; and
 - n. Management Tools enabling the support of the Management Applications Requirements detailed in section [9.7.2](#) of this document.

9.5 Server Applications

The DND defines an M&S/SE Server Application as an application that should provide a certain set of M&S/SE services to the DND user community, whereas these Server applications can be from fully “Centralized” to fully “Distributed” or Net-centric.

In a Client/Server model as stated previously, Clients connect to a Simulation Domain and/or single and/or many Servers that should provide all Simulation Services to the Client Applications. The Concept of a Simulation Domain or Server can be compared to Windows Domain in the Windows architecture and also to the following Servers’ concept:

- a. Networking;
- b. File;
- c. Print;
- d. Security;
- e. Access Control;
- f. Database;
- g. Email; and
- h. Web.

In an M&S/SE concept, Centralized Server(s) should offer a certain set of services to clients connected to the network. These Servers can work together to form a single Simulation Domain or be segregated into their own domains.

9.5.1 In the future M&S/SE world, the type of services required from a Simulation Server should be as follows:

- a. Weather;
- b. Scoring;
- c. Terrain;
- d. Communications;
- e. Dynamics [It refers to physics and movements or information dynamics of the simulation environments themselves];

- f. Access Control;
- g. Expert System;
- h. Security;
- i. Positioning;
- j. Navigation; and
- k. Sensors.

These Server Applications should provide all Simulation Services required for the Clients Applications that perform different sets of processes.

It is important to note that these Simulation Services should run on top of standard Operating Systems, such as Windows, in providing the complete set of required services.

A vendor should offer to DND over time multiple new Server applications that provide different type of features and functionalities in support of M&S/SE requirements as per the aforementioned list.

9.5.2 First and foremost, the key Server application required is a Dynamic Synthetic Environment (DSE)/Computer Generated Forces (CGF; described in more details in Section 9.9) that should provide a full Simulation Environment for the following:

- a. Training for the following:
 - i. Provides interactive Networked/Distributed tactical environments to trainee for a full range of civilian or military operations; and
 - ii. Simulates dangerous and costly missions.
- b. Research and Development (R&D) as follows:
 - i. Study effectiveness of advanced platforms and on-board systems; and
 - ii. Develop tactics and doctrine.
- c. Mission Planning, Awareness and Rehearsal as follows:
 - i. Tactics, techniques, and procedures against threat; and
 - ii. Validation of new Platforms.
- d. Education as follows:
 - i. Supplies a platform for study projects.
- e. Support the ‘Generic Usage’ statement in paragraph “h” of section [9.1.1](#) of this document;
- f. Enable to support and offer Simulation Services to the “DND User Community” as per the statement in section [9.4.1](#) of this document; and
- g. Enabled or architected to support the capability of being fully centralized, decentralized, fully distributed, or mixed-architecture in support of the modular conceptual M&S/SE framework.

9.5.3 Support for M&S/SE Concepts and Attributes

These Server Applications should support the Concepts and Attributes of the M&S/SE framework detailed in section [9.1.1](#) of this document as follows:

- a. Physical/Data Link/Network/Transport;
- b. Enabling Legacy Applications;
- c. Open Architecture;
- d. Open Programming Structure;
- e. Interoperable;
- f. Cross-Platform;
- g. Scalable;
- h. Generic Usage;
- i. Extensible;
- j. Human Oriented Design;
- k. Licensing Mechanism;
- l. Software Good Distribution Methods;
- m. Upgradeability;
- n. Provision to implement Security and Access Control;
- o. Network Centric;
- p. Multiple Language Support; and
- q. The Server Applications should also support the following:
 - i. Load-Balancing/Sharing technologies i.e. enable to support sharing or load on multiple systems or processors to ensure scalability;
 - ii. Provisions for Redundancy as follows:
 - 1) Provisions to be enabled to support Redundancy technologies to ensure continuous operations; and
 - 2) Failover protocols and technologies to ensure that redundancy mechanisms are properly used and can be post-failure debugged.

9.5.4 Generic Server Applications additional requirements should be as follows:

- a. The Professional Documentation should cover the following subjects:
 - i. Introduction;
 - ii. Acronyms and Terminology;
 - iii. Quick References;
 - iv. Frequently Asked Questions;
 - v. System Requirements;
 - vi. Network Requirements;
 - vii. Infrastructure Requirements;
 - viii. Generic OS Requirements;
 - ix. RTI Requirements;
 - x. Product Installation;
 - xi. Getting Started;
 - xii. Planning and Configuration;
 - xiii. Operation;
 - xiv. Management;
 - xv. Administration;

- xvi. Debugging;
- xvii. User Interface; and
- xviii. Known Problems.

- b. Management Tools enabling the support of the Management Applications Requirements are detailed in section [9.7.2](#) of this document.

9.6 Distributed HLA Applications

The Distributed or Networked-centric Applications should be High-fidelity specific applications that require their own dedicated platform or system for operation e.g. Weather Server providing the same weather patterns to multiple simulations or the same Weather Server providing multiple different weather patterns to each simulation, via a networking technology.

It should offer the possibility to DND Developers or DND Systems Architects to leverage specific capabilities or expertise of systems or specific areas within the DND or its partners/vendors.

In special cases, specific vertical and/or horizontal applications will be required by participants. This could be a military application that requires a higher level of fidelity or Service specific applications. These types of applications should be also used to support the DND M&S/SE framework initiative, by enabling fully distributed Simulation Applications to interoperate over a networking technology such as Local or Wide-area Networking (LAN or WAN).

In the future these Distributed or Networked-centric Applications should become the foundation for the implementation of a modular conceptual M&S/SE framework at DND. This should enable the development of modular and innovative new applications that should foster better commonality and interoperability; hence resulting into greater reusability and cooperation between DND Entities.

9.6.1 Support for M&S/SE Concepts and Attributes

These Server Applications should support the Concepts and Attributes of the M&S/SE framework detailed in section [9.1.1](#) of this document as follows:

- a. Physical/Data Link/Network/Transport;
- b. Enabling Legacy Applications;
- c. Open Architecture;
- d. Open Programming Structure;
- e. Interoperable;
- f. Cross-Platform;
- g. Scalable;
- h. Generic Usage;
- i. Extensible;
- j. Human Oriented Design;

- k. Licensing Mechanism;
- l. Software Good Distribution Methods;
- m. Upgradeability;
- n. Provision to implement Security and Access Control;
- o. Network Centric; and
- p. Multiple Language Support.

9.6.2 Distributed HLA Applications additional requirements should be as follows:

- a. Enable to support future Distributed HLA Application Requirements as follows:
 - i. Hi-fidelity Weather Server;
 - ii. Human Behavioural Representation (HBR) Server;
 - iii. Dynamic Terrain Server;
 - iv. Sensor Server;
 - v. Communications Server;
 - vi. C4ISR Server; and
 - vii. After Action Review (AAR) Analysis Server.
- b. The Professional Documentation should cover the following subjects:
 - i. Introduction;
 - i. Acronyms and Terminology;
 - ii. Quick References;
 - iii. System Requirements;
 - iv. Network Requirements;
 - v. Infrastructure Requirements;
 - vi. Generic OS Requirements;
 - vii. RTI Requirements;
 - viii. Product Installation;
 - ix. Getting Started;
 - x. Configuration;
 - xi. Operation;
 - xii. Management;
 - xiii. Administration;
 - xiv. Debugging;
 - xv. User Interface; and
 - xvi. Known Problems.
- c. Management Tools enabling the support of the Management Applications Requirements detailed in section [9.7.2](#) of this document.

9.7 Management Applications

In a concept of Client/Server and/or Networked-centric/Distributed Simulation Environment, Management and Administration of the Components are undoubtedly becoming an essential factor for DND and GOC. There is a clear parallel to the Networking Industry with the

creation of Distributed LANs in the late 80's, where its administrator was called the "Network Administrator". In the Distributed Simulation environment, a new breed of Administrator has evolved called the "Simulation and/or Scenarios Manager", which is related but still very different to the Networking Industry. The new category of personnel must manage advanced knowledge-based concepts as well as the technical concepts and to do this, advanced tools must support them.

In the early phases, the M&S/SE Industry should provide Simple Management Tools to control basic sets of Features and Functionalities. However, as the complexity of the Simulation Environments evolves, new requirements will emerge such as Simulation Asset (Computing, Devices, and Networking) Management, Multi-Level Security (MLS), Cross Scenario Communication, LAN and WAN Interconnection, Access Rights, IP rights, and Ownerships.

Simulation Managers will become an important part of the Networking/Computing Administration Teams in large organizations leveraging M&S/SE for their business needs. Initially, the current Networking/IT/Application Administrators will probably fill the role of Simulation Managers.

Current Management Applications should offer basic Management Functionalities to control the Server components of the Simulation, some of the Simulation Environment Variables, and present information on the Entities and the Simulation.

As the Complexity of the Tools and the Simulation Environments increases and become wider ranging, a Simulation Management Application which is a fully Centralized/Decentralized Console will be required.

The Management Applications will become in itself their own domain providing a complete Vertical Application Layer with its own set of APIs and end-user applications in order to provide management capabilities to Simulation Managers in conjunction with IT Managers within organizations.

9.7.1 Support for M&S/SE Concepts and Attributes

These Server Applications should support the Concepts and Attributes of the M&S/SE framework detailed in section [9.1.1](#) of this document for the following:

- a. Physical/Data Link/Network/Transport;
- b. Enabling Legacy Applications;
- c. Open Architecture;
- d. Open Programming Structure;
- e. Interoperable;
- f. Cross-Platform;
- g. Scalable;
- h. Generic Usage;
- i. Extensible;
- j. Human Oriented Design;
- k. Licensing Mechanism;

- l. Software Good Distribution Methods;
- m. Upgradeability;
- n. Provision to implement Security and Access Control;
- o. Network Centric; and
- p. Multiple Language Support.

9.7.2 Management Applications should be capable of providing the following set of services:

- a. Security;
- b. Encryption;
- c. Access Control & Rights;
- d. Granular Entity Management;
- e. Aggregate Management;
- f. Warlord Centralized Controls;
- g. Instructor and Training Requirements;
- h. Simulation Environment Controls;
- i. Environment Management;
- j. Rewind, Fast-Forward, Playback;
- k. Cross Scenario Communication;
- l. LAN/WAN Performance Monitoring;
- m. Distributed Monitoring;
- n. Redundancy;
- o. HLA Application Distribution and Registration; and
- p. Networking Management Station Integration.

9.7.3 The Professional Documentation should cover the following subjects:

- a. Introduction;
- b. Acronyms and Terminology;
- c. Quick References;
- d. System Requirements;
- e. Infrastructure Requirements;
- f. Generic OS Requirements;
- g. RTI Requirements;
- h. Product Installation;
- i. Getting Started;
- j. Configuration Management;
- k. Operation Management;
- l. Fault Management;
- m. Resource Management;
- n. Application Management;
- o. Scenario Management;
- p. Time Management;
- q. Model Management;
- r. Event Management;
- s. Event Logging;
- t. Administration Services;

- u. Debugging;
- v. Problem Solving;
- w. User Interface; and
- x. Third-Party Product Management.

9.8 Common Synthetic Environment (CSE) Infrastructure

The Common Synthetic Environment (CSE) infrastructure should provide a generic implementation of the services provided by the Common Simulation Services (CSS). Any Goods procured must be multi-disciplinary in use and they should provide generic use to the DND community and beyond. They should also provide a Common Synthetic Environment (CSE) infrastructure capability to ensure the viability of following concepts:

- a. Reusability;
- b. Commonality; and
- c. Interoperability.

The vendor solution should support the concept of a General Common Synthetic Environment (CSE) infrastructure to enable all DND Stakeholders and Users to have a minimum set of M&S/SE requirements met.

DND does not expect that the CSE infrastructure will solve all requirements for all users. However, the definition of this layer should enable for a long-term commonality throughout all solutions developed or offered within DND by addressing a certain percentage (%) of all DND M&S/SE requirements.

9.8.1 Support for M&S/SE Concepts and Attributes

This Common Synthetic Environment infrastructure should support the Concepts and Attributes of the M&S/SE framework detailed in section [9.1.1](#) of this document for the following:

- a. Physical/Data Link/Network/Transport;
- b. Enabling Legacy Applications;
- c. Open Architecture;
- d. Open Programming Structure;
- e. Interoperable;
- f. Cross-Platform;
- g. Scalable;
- h. Generic Usage;
- i. Extensible;
- j. Human Oriented Design;
- k. Licensing Mechanism;
- l. Software Good Distribution Methods;
- m. Upgradeability;
- n. Provision to implement Security and Access Control;
- o. Network Centric; and

- p. Multiple Language Support.

9.8.2 The Common Synthetic Environment (CSE) infrastructure should also include the following requirements:

- a. Professional Documentation required as follows:
 - i. Introduction to the Common Synthetic Environment (CSE) infrastructure;
 - ii. High-level concepts of the CSE infrastructure;
 - iii. How to expand the CSE Infrastructure;
 - iv. Common Repository Management;
 - v. Features and Functionalities Management;
 - vi. CSE infrastructure main components;
 - vii. Services provided by the CSE infrastructure;
 - viii. Commonality;
 - ix. Reusability;
 - x. Security;
 - xi. Access Control;
 - xii. Database Management;
 - xiii. Object Profiling;
 - xiv. Verification;
 - xv. Validation;
 - xvi. Analysis;
 - xvii. Accreditation,
 - xviii. Certification, and
 - xix. Upgrade/Updates.
- b. FOM agility component that should be available from the simulation runtime to isolate the application from the FOM used on the distributed simulation network. The FOM agility component should map the applications API to the selected FOM;
- c. Terrain Management should provide the following:
 - i. Terrain database management as follows:
 - 1) Loading of a terrain database;
 - 2) Terrain Paging; and
 - 3) Database compatible sources as follows:
 - i. Open Flight,
 - ii. Terra Page;
 - iii. DTED;
 - iv. DFAD; and
 - v. J2-GICI Compatible formats.
 - ii. Line of sight calculations (LOS);
 - iii. Height above terrain (HAT);
 - iv. Terrain Inclination for vehicle clamping; and
 - v. Support for Polygonal and terrain grid format.
- d. Positioning as follows:
 - i. Position approximation/extrapolation; and
 - ii. The systems implement the SEDRIS position keeping libraries.

- e. Weather components should provide a data structure for the definition and exchange of the following weather parameters:
 - i. A Wind vector in three dimensional space; and
 - ii. Atmospheric pressure and temperature.
- f. The following navigation parameter should be definable for entities as follows:
 - i. Position;
 - ii. Orientation in space;
 - iii. Velocity vector;
 - iv. Acceleration vector;
 - v. Route defined by waypoints as follows:
 - 1) Position;
 - 2) Speed; and
 - 3) Estimated Time of Arrival (ETA).
 - vi. The navigation parameters are modifiable in run time by direct user inputs as follows:
 - 1) Manual I/O device; and
 - 2) Client software.
- g. Provide a means to detect collision between entities/terrain/structures;
- h. Sensors and situation awareness as follows:
 - i. Provide a data structure to store entities perception of its environment;
 - ii. Acquire the environment perception from entity sensory systems; and
 - iii. Provide a means to detect and transmit emissions.
- i. Defensive Systems or Countermeasures as follows:
 - i. Provide the capability of disrupting the sensor operation of other entities present in the synthetic environment; and
 - ii. Provide a means to disrupt the operation of another entity in the synthetic environment.
- j. Communications should provide the following:
 - i. Provide a mechanism to store and exchange information between entities.
- k. Aggregation should provide the following:
 - i. Allows a large number of entities to be integrated as a single entity;
 - ii. Allow for aggregation and de-aggregation during the execution;
 - iii. Support the fully aggregated mode wherein a group of entities is simulated as one; and
 - iv. Support a partially aggregated mode wherein a group of entities is simulated as one but each of the entities is still visible and can still interact with the other entities.
- l. Data Management as follows:
 - i. Scenario Definitions;
 - ii. Entity Definitions;
 - iii. Equipment definition;

- iv. Rule database and/or Rule-base management; and
 - v. Knowledge-base management.
- m. Behavioural Systems should support the following:
- i. Elements of the simulation to be controlled through a set of behavioural rules;
 - ii. Simulation elements that may have behavioural rules include entities, aggregate entities and equipments;
 - iii. Set of rules to be assigned to individual simulation elements; and
 - iv. Set of rules to be assigned to a group of simulation elements.
- n. Collection and processing of data to be used for the computation of metrics to allow evaluation of modeled performance within the simulated scenario;
- o. Scoring system to maintain a consistent health status for each simulation entity;
- p. Analysis Tools to support the following:
- i. Briefing;
 - ii. Debriefing;
 - iii. Post Analysis; and
 - iv. Reporting.
- q. Management Tools enabling the support of the Management Applications Requirements detailed in section [9.7.2](#) of this document.

9.9 Dynamic Synthetic Environment & Computer Generated Forces (DSE/CGF)

The Dynamic Synthetic Environments & Computer Generated Forces (DSE/CGF) requirements should be as follows:

- a. Provide a high-fidelity synthetic environment for air, land, and sea scenarios through a series of application libraries;
- b. The Application framework defining the object models from components that conform to the RPR-FOM v1.0 or v2.0 structure and attribute list to ensure maximum compatibility with existing standards;
- c. The DSE/CGF package offered with different levels of Fidelity from Simple physical interactions to Full Reality Operation Rehearsal;
- d. Provide libraries of Low, Medium, and High fidelity models;
- e. Includes an integrated graphical user interface (GUI);
- f. Capable to be used out-of-the-box as a fully-functional DSE/CGF application;

- g. The DSE/CGF applications support a Client/Server model in order to distribute its control over a network;
- h. The DSE/CGF simulate a real-world environment that include the following:
 - i. Dynamic interaction between entities;
 - ii. The mathematical models that are physics-based models considering the principal parameters affecting entity behaviour/performance and evolution in the dynamic environment; and
 - iii. Entities characteristics accurately integrated to provide high fidelity sensor stimulation.
- i. The DSE/CGF should enable the Operator staff to create different scenarios operations for the following:
 - i. Naval Operations;
 - ii. Emergency Management Services;
 - iii. 911 Police Management;
 - iv. Ground Operations;
 - v. Air Campaigns;
 - vi. Research and Development;
 - vii. Air Traffic Control Simulations;
 - viii. Urban Planning;
 - ix. Urban Combat;
 - x. Chemical Biological Radiological, and Nuclear (CBRN) Evaluations;
 - xi. Military Mission Rehearsal;
 - xii. Civilian Mission Rehearsal;
 - xiii. Search And Rescue (SAR);
 - xiv. Operational Analysis; and
 - xv. Combination of the above.
- j. The DSE/CGF architecture supports different levels of fidelity as required by each application e.g. an air combat trainer requires high level of fidelity flight model for the entities operating close to the virtual simulator whereas an Air-Traffic control flow model requires relatively low individual model fidelity;
- k. The DSE/CGF should be able to create a realistically simulated multi-entity type, multi-platform, time-stressed environment comprising of the following:
 - i. Group of entities operating in competitive or friendly teams within a gaming area;
 - ii. Entities with respective dynamics (velocity and acceleration), signatures (detectable by the entity sensors e.g. electro-optical and radar), vulnerability, equipment (sensors, countermeasures, on-board systems, communication devices, and payload); and
 - iii. Entities interact with live, virtual and constructive models, and according to their dictated behaviour.
- l. Platforms available to the DSE/CGF Operator or Instructor to build and operate in a real time scenario. The platforms are generic entity structures which may be used to build and save specific entities for use in scenarios;

- m. Customization of the platform is accomplished by the parameterization of entity dynamics, geometry, operating limits and modes of operation;
- n. The Generic Entity Type Support should be as follows:
 - i. Aircraft category as follows:
 - 1) Fighter/Attack;
 - 2) Bomber;
 - 3) MPA;
 - 4) Reconnaissance; and
 - 5) Transport.
 - ii. Rotorcrafts category as follows:
 - 1) Attack;
 - 2) ASW;
 - 3) Scout;
 - 4) Utility; and
 - 5) UAV (Unmanned/Uninhabited Air Vehicle).
 - iii. Track Vehicle category as follows:
 - 1) SAM;
 - 2) ADA;
 - 3) Heavy Tank;
 - 4) Medium Tank;
 - 5) Light Tank;
 - 6) Armoured Personnel Carrier;
 - 7) Bulldozers;
 - 8) Snowmobiles; and
 - 9) Civil tracked vehicle.
 - iv. Non-Track Vehicle category as follows:
 - 1) Truck;
 - 2) Car;
 - 3) Trains;
 - 4) Ambulance;
 - 5) Police Cruisers;
 - 6) Jeep; and
 - 7) Armoured Personnel Carrier.
 - v. Fixed Ground Category as follows:
 - 1) Buildings;
 - 2) Bridges;
 - 3) Towers;
 - 4) Power plants;
 - 5) Dams;
 - 6) Radar Station;
 - 7) SAM; and
 - 8) Industrial Complexes.
 - vi. Ships Category as follows:
 - 1) Naval vessel as follows:
 - a. Aircraft Carrier;
 - b. Cruiser;

- c. Destroyer;
 - d. Frigate;
 - e. Patrol boat; and
 - f. Replenishment.
 - 2) Freighter;
 - 3) Tanker;
 - 4) Container Ship; and
 - 5) Fishing Trawler.
 - vii. Life-raft Category as follows:
 - 1) Single-seat; and
 - 2) Multi-seat.
 - viii. Fixed Surface Category as follows:
 - 1) Buoys; and
 - 2) Oil Rigs.
 - ix. Organisms as follows:
 - 1) Humans;
 - 2) Animals; and
 - 3) Plants.
 - x. Subsurface Category as follows:
 - 1) Submarine; and
 - 2) Wreck.
- o. Database Management as follows:
- a. Scenario Definitions;
 - b. Entity Definitions;
 - c. Equipment definition; and
 - d. Rule database.
- p. Weapons required should be as follows:
- i. The ballistic model (i.e. gun rounds, rockets, and depth charges) is a model which considers drag and gravity drop;
 - ii. The rocket model considers the thrust developed by the propulsion system;
 - iii. Missiles and torpedoes exhibit dynamic and behavioural characteristics appropriate to the type of guidance system and sensory target acquisition capabilities, in particular their susceptibility to countermeasures;
 - iv. The effectiveness of weapon systems should be computed based on actual weapon performance including the following factors:
 - 1) Trajectory;
 - 2) Accuracy;
 - 3) Dispersion;
 - 4) Effective and maximum range; and
 - 5) Realistic weapon conditions.
 - v. Weapons Management system required should be as follows:
 - 1) Capable to receive commands from a rule based system;
 - 2) Capable to receive commands from a user interface;

- 3) Allow the Expert system to select the weapon class to be fired. (e.g. seeker type);
 - 4) Allow the Expert system to select the weapon type to be fired. (e.g. gun, rocket, bomb, and missile); and
 - 5) Allow the Expert system to select a weapon by station.
- q. Defensive Aid System (DAS) required should be as follows:
- i. The DAS of an entity uses this entity awareness of the current battlefield environment to activate, deactivate, release and stop releasing countermeasure expendable;
 - ii. Be "connected" to the Sensor-Suite to emulate real platform "Survival Suite";
 - iii. Receive command input from the Expert-System;
 - iv. Receive command input from the user interface in order to override automatic or expert system driven behaviour;
 - v. Capable to receive predefined command as standard response to Sensor-Suite;
 - vi. The following countermeasures supported should be as follows:
 - 1) Armour;
 - 2) Chaff;
 - 3) IR Flare;
 - 4) RF Jammer;
 - 5) IR Jammer;
 - 6) Laser Jammer;
 - 7) Radio Jammer;
 - 8) Smoke Generator; and
 - 9) Tactical Smoke.
- r. Sensors required should be as follows:
- i. Emulate the environment perception of each simulated entities in a scenario. The sensors' perception should be consistent as follows:
 - 1) Amongst simulated entities;
 - 2) With virtual entities;
 - 3) With live entities; and
 - 4) Representative emissions are generated from active sensors.
 - ii. Sensor models should fulfill the following requirements:
 - 1) Locate entities present in the world;
 - 2) Compute and update track position; and
 - 3) Classification of detected entity base on signature, behaviour and other scenario information.
 - iii. Sensor simulations should take into account the following:
 - 1) Terrain obscuration;
 - 2) Back-scattering;
 - 3) Presence of countermeasures.
 - 4) Environmental effects; and
 - 5) Time of day.

- iv. The Entity Signature model based on physical characteristics;
- v. Sensor Types required should be as follows:
 - 1) RF;
 - 2) Magnetic;
 - 3) Acoustic;
 - 4) Warning Receivers;
 - 5) IFF Interrogator/Transponder;
 - 6) Electro-Optical Systems; and
 - 7) Laser System as follows:
 - a) Laser Designation;
 - b) Range-Finding;
 - c) Beam-Rider; and
 - d) Tracker functionality.
- s. Situation Awareness required should be as follows:
 - i. The DSE/CGF has a means to process and fuse the information from different sensor sources into a consolidated contact pictures for the expert system;
 - ii. Contact sources accept datalinks contacts as well as sensor tracks;
 - iii. Track management includes mechanisms to handle lost of contact and status change of contact such that contacts remain or are deleted from the track list used by the expert system; and
 - iv. The track management system has the possibility to memorize and interpolate track positions that have lost sensor contacts.
- t. Dynamics required should be as follows:
 - i. The Dynamics subsystem emulates the forces and moments acting on an entity for all CGF entity types. The level of fidelity (LOF) of the dynamics simulation is selectable by the user as follows:
 - 1) Lower fidelity modeling allows the operation of a high number of entities on a single Computer [Upwards of at minimum 500 entities to an unlimited number if possible. This is an area that is currently not well-specified in the “international” community since most use polygon count for scene complexity and the metrics are different and still evolving for simulation entities.];
 - 2) A higher level of fidelity is accessible for a selected number of entities to allow for realistic dynamic behaviour; and
 - 3) Simple dynamics should consist of the following:
 - a. A behaviour that is consistent with its environment (e.g. ground attitude for ground entities);
 - b. Visually representative attitude that represent the entity actions (e.g. an helicopter that accelerates shall pitch to accelerate); and
 - c. Based on its dynamic envelope (min/max speed, acceleration, pitch and roll angles, and rates).
 - ii. The entity dynamics driven by inputs from the navigations system and the DSE/CGF operator. The operator has enough manual control available to navigate the entities;

- iii. The entity dynamics parameterized such that the user can change and create specific vehicle e.g. from a generic aircraft dynamics a user is able to insert a B-737 by entering its physical parameters through the user interface;
- iv. Dynamics Models required should include the following:
 - 1) Simple (Player) Dynamics as follows:
 - a. Aircraft;
 - b. Rotorcraft;
 - c. Naval Vehicle;
 - d. Ground Vehicle;
 - e. Sub-surface Vehicle;
 - f. life-forms; and
 - g. Spacecraft.
 - 2) Enhanced Dynamics - Aircraft. The forces and moments computed taken into consideration should be as follows:
 - a. Gravity;
 - b. Lift forces and moments;
 - c. Drag forces and moments;
 - d. Ground forces such as landing gear and brake;
 - e. Engine Thrust; and
 - f. Change in surfaces.
 - 3) Enhanced Dynamics - Rotorcraft. It should be based on rotor disk model (or higher fidelity) including the following:
 - a. Gravity;
 - b. Lift forces and moments;
 - c. Drag forces and moments;
 - d. Ground forces (such as landing gear and skids) and moments;
 - e. Engine Thrust and moments;
 - f. Rotor Thrust and moments; and
 - g. Change in rotor disk and tail rotor.
 - 4) Enhanced Dynamics - Naval Vehicle and Sub-surface Vehicle requirements should be as follows:
 - a. The Forces computed should be as follows:
 - i. Buoyancy force;
 - ii. Aerodynamics force;
 - iii. Hydrodynamics force;
 - iv. Thrust;
 - v. Gravity force; and
 - vi. Anchor force.
 - b. The Moments computed should be as follows:
 - i. Moment due to rudder deflection;
 - ii. Moment due to variation of the center of buoyancy;
 - iii. Moment due to aerodynamics force; and
 - iv. Moment due to hydrodynamics force.
 - 5) Enhanced Dynamics – Ground Vehicle requirements should be as follows:
 - a. Compute dynamics system degradation due to ground conditions or environment conditions; and

- b. Ground vehicle takes into account impact forces on other entities or features.
- u. Navigation/Manoeuvring requirements should be as follows:
 - i. Control how a player achieves its objective e.g. it ensures that the entity proceeds towards position while at the same time following and avoiding collision with the terrain;
 - ii. Interface primarily with the expert system, dynamics and terrain systems. This system receives the Manoeuvre Mode commands from the Rules, processes these commands in order to navigate the entities through the terrain area by sending Manoeuvre Control requests to the Dynamics;
 - iii. Navigation Command Types should be as follows:
 - 1) Speed Command: It should be possible for entities to accept/perform speed change commands for the following:
 - a. Ground Speed;
 - b. True Air Speed (TAS); and
 - c. Indicated Air Speed (IAS).
 - 2) Altitude Command: It should be possible for entities to accept/perform altitude change commands for the following:
 - a. Above (Mean) Sea Level (MSL);
 - b. Above Ground Level (AGL); and
 - c. Pressure Altitude.
 - 3) Heading Command: It should be possible for entities to accept/perform heading change commands for the following:
 - a. Ground Track control;
 - b. Heading control;
 - c. Magnetic heading control; and
 - d. Magnetic ground track control.
 - 4) It is possible to control an entity in all its Degrees-of-Freedom (DOF);
 - 5) Waypoints Command. It is possible for entities to navigate to a geographical coordinate in latitude, longitude and altitude (or other co-ordinate system);
 - 6) Waypoints are collection of geographical coordinates and require the following:
 - a. Possible to achieve a given geographical coordinate at a specified time; and
 - b. Possible to achieve a given geographical coordinate at a specified speed.
 - 7) The navigation modes to be supported as a baseline should be as follows:
 - a. Nap of the Earth (NOE): Navigation mode in which different paths are weighted for the entity. The path that lets the player travel at an almost constant speed, altitude and with large lateral movements and minimum exposure is chosen. Obstacle avoidance is performed in the XY-axis;
 - b. Direct;
 - c. Contour: With this method obstacles are located a short time before reaching they are reached and avoidance is done in the Z-axis. The objective is to keep a relatively constant ground elevation; and

- d. Low level: Same as contour except that obstacle avoidance is done much ahead, ground elevation is allowed to vary.
- v. Collision Avoidance requirements should be as follows:
 - i. Possible for entities to avoid collisions with other entities or features.
- w. Joystick inputs used to control a player should be as follows:
 - i. X/Y;
 - ii. Pitch;
 - iii. Roll;
 - iv. Yaw; and
 - v. Throttle.
- x. Physics Based Scoring requirement using:
 - i. Damage producing levels; and
 - ii. Zone endurance.
- y. Hierarchical Structure requirements should be as follows:
 - i. Hierarchical structure should support these different types of organization as follows:
 - 1) Hierarchy Organization/Structure i.e. Command relationship as follows:
 - a. Represents organizations (group and sub-group) from the point of view of Command relationship; and
 - b. Organizes all the hierarchical structures in term of commanded groups.
 - 2) Formation i.e. Manoeuvre and Spatial (geographic position) relationship as follows:
 - a. Represent geometrical position of the formation (formation style);
 - b. Specify the master of the formation as navigation lead; and
 - c. Accept different type of players as follows:
 - i. Possible for different player types to be in the same formation;
 - ii. Possible to modify a formation at run time;
 - iii. Possible to modify formation style;
 - iv. Possible to do a fast formation change;
 - v. Formation handling includes generic approaches to formation changes;
 - vi. Formation handling includes generic approaches to formation turn;
 - vii. Possible to define a formation as an aggregation of formations (super-formations); and
 - viii. Possible to break-off or ungroup a formation at run-time.
 - 3) Composition i.e. Spatial (geographic position) relationship as follows:
 - a. The DSE/CGF supports composition (Entity "within" another Entity e.g. example used to represent Aircraft carrier containing aircrafts); and
 - b. A composition is dynamically modified by adding or removing a member to it e.g. when a player is landed on a ship.
 - 4) Communication Network i.e. Communication relationship.

- ii. Zoning requirements should be as follows:
 - 1) Points, lines, and zones are used in the virtual area to define behaviours based on geographical locations and should include the following:
 - a. Possible to define points, lines and areas on the GUI that can be used by the expert system;
 - b. The points, lines and zones appear on the GUI as overlays;
 - c. Area can have different types (no-tack Joint SAR, asset, objective, and FARPS) e.g. The behaviour of a player inside an area has to be described as a set of rules;
 - d. Possible to refer to an area in order to carry out command;
 - e. Possible to define different shapes of areas i.e. Rectangle, circle, sector, triangle and ellipse;
 - f. Possible to specify a speed for a moving area;
 - g. Possible to define team ID and name for a zone;
 - h. Possible to determine if the player is inside a specific area, passed a point or a line, in order to carry out commands.
- iii. Entity Aggregates requirements should be as follow:
 - 1) Constructive simulation or aggregate simulation control groups of entities as an aggregate rather than as a set of individual simulated entities;
 - 2) Emerging aggregate properties allow individual entities to dynamically combine or split properties and/or capabilities;
 - 3) Capability to dynamically change the level of aggregation;
 - 4) Support mechanisms of de-aggregation based on the following:
 - a. Fixed geographical area;
 - b. Manual triggering;
 - c. Sphere of influence; and/or
 - d. Events.
 - 5) Support multiple levels of aggregation. That is, it is possible to support aggregation of aggregations; and
 - 6) Possible to define aggregation geometry based on simple geometrical primitives e.g. line and arc circle.
- iv. Communications System requirements should be as follows:
 - 1) Based on physical system characteristics; and
 - 2) The DSE/CGF communications model simulates an RF based communication system;
- v. Expert System requirements should be as follows:
 - 1) Entity behaviour controlled by an expert system;
 - 2) The expert system rules is user definable;
 - 3) A set of readily available parameters assign to types of behaviour is selectable from the user;
 - 4) Rules are stored in a database;
 - 5) Rules are assigned to entities at scenario creation or run time;
 - 6) Behaviour is assignable to DSE equipment, entities, and aggregate

- 7) Expert system supporting a distributed architecture and mode of operation;
 - 8) Expert system user interface supporting non- programmers;
 - 9) An advanced mode supported for programmers and specialists; and
 - 10) The rules system capable to handle large rule sets (measurable in the thousands and not just hundreds of rules).
- z. Management Tools enabling the support of the Management Applications Requirements detailed in section [9.7.2](#) of this document.

10. Services Requirements

DND has learned over years of experience that the support, education/training, and professional services accompanying software is as important, if not more, than the software suite per se. We felt it was important to document in sufficient details the generic requirements for Services that are part and parcel of any software solutions if DND wants that holistic solution to be readily available, affordable, generally accepted, consistent and harmonious with other Canadian and/or abroad M&S/SE users.

A synergistic approach that serves the best needs of DND is based upon the fact that applications and orthogonal services, provide the greatest opportunity for positive partnering. However, divided interests from fragmented requirements do not serve the industry or the government well at all. Considering the size of the M&S/SE market in Canada, considering there is still no critical mass in M&S yet, DND will need to look at potential coalition and interoperability opportunities stemming from a state-of-the-art “requirements” based approach in order to achieve economy of scope and scale.

10.1 Support Services

The M&S/SE Support services would accomplish the following in supporting the modular M&S/SE framework:

- a. More flexible and adaptive mechanisms and methods for integrating disparate existing software applications;
- b. Improved ability to reflect the dynamics of evolving methodologies and systems, resource uses, and capabilities management practices;
- c. Capability to support software applications that can operate at multiple political, social, spatial and temporal scales; and
- d. Reduction in the long-term cost of modeling technology by use and reuse of existing data, models, and system components through a coherent support services base as well as reduce the development time required before one can actually use this simulation capability.

The Support Services comprise of a specific set of after-market services to support the customer’s R&D development, experiment, deployment and operational needs.

10.1.1 Support Services Requirements

The Requirements for Support Services in support of the modular Modeling and Simulation/Synthetic Environments (M&S/SE) framework should be as follows:

- a. Telephone;
- b. Web;
- c. Email;

- d. On-Site [It is performed at the DND user's site but do not include Educational and Training Services, but may include limited Professional Services;
- e. Remote [Using technologies that enable quick response to problems and issues and that limit the amount of On-Site Support Services required];
- f. Warranty & Maintenance;
- g. Customers Messaging System [System that enables the vendor to provide, via a communication channel, product Support information to customers];
- h. Trouble Ticket System;
- i. File Transfer Protocol (FTP) site [It is an Internet Site such as an IP address or a dedicated name (company.ftp.com) that provides a set of internet download services based on FTP. This FTP site should be accessible using standard Web browsing technologies or specialized FTP software];
- j. Account Management [It is defined as a set of dedicated offerings for large DND user based upon whether the user community is a single department or Branch or an entire DND entity];
- k. Escalation Process Support [The vendor's Support Services Infrastructure should have the capability to support different escalation paths depending on the Severity Level of the problem/issue or specific Service Level Agreement (SLA) requirements offered to DND users];
- l. Software Design/Development/Release Cycles [The vendor should have processes that clearly identify the COTS Software Release Cycle as part of the standard Product Life Cycle];
- m. Configuration Management (CM);
- n. Quality Assurance (QA);
- o. Features and Functionalities Request Process [The vendor should have Features and Functionalities (FF) Request Policies & Procedures (P&P) in place to facilitate the users into the utilization of the FF Request process. As a result, DND users will create a "push" demand on the FF of the M&S/SE framework therefore assist the vendor into prioritizing and acting on the needs of DND;
- p. Defective Goods Return Policy [The vendor should have Defective Goods Return Policies & Procedures (Return Materials Authorization (RMA) process) in place for Goods under warranty (returned to the vendor's facility) or other Support Service coverage services back to the original vendor (OEM) in order to ensure proper replacement of the defective part or proper credit of the aforementioned.;
- q. Goods End-of-Life (EOL) Policy [The vendor should support Goods offered through the life and beyond of a defined contract agreement];
- r. Service Level Agreement (SLA);
- s. Bug tracking/Reporting System; and
- t. Usage Tracking/Reporting System.

10.2 Educational & Training Services

Currently, M&S/SE Education/Training services are ad hoc, fragmented, disjointed, performed haphazardly, and repetitive, in the sense that many firms will perform introductory level of M&S/SE course but few, if any, will regularly perform intermediary or advanced

levels of M&S/SE courses, thus offering a stagnant level of information, data and knowledge to the community and thus not empowering the community to rise to the next level.

Based on the modular M&S/SE framework, there is a need for Education and Training both at the Entry and Advanced levels. The courses that would fall under “Educational” would be generic courses that would be required to broaden and heighten the knowledge and awareness of M&S/SE among the DND user community and also to prepare the users prior taking specific courses (from the vendor or Third-parties) on simulation tools or environments that fall under “Training”.

10.2.1 DND Members Benefiting from M&S/SE Educational & Training Courses

It is clear that the same categories of members of the DND community identified in Section 9.4.1 could benefit from the Educational and Training courses described below.

10.2.2 Specific Personnel Categories Requiring Specific M&S/SE Educational & Training Courses

The type of personnel that would not only benefit but require knowledge on M&S/SE Concepts, Processes and Technologies are divided in four categories as follows:

- a. User/Operator is described as follows:
 - i. Operator - Personnel required to operate a Simulation System Operating Station to either train personnel or support a user;
 - ii. Element Designer - Personnel required to create new entities or objects for a scenario or simulations;
 - iii. Scenario/Simulation Designer - Personnel required to enter complex scenarios or to create full simulations in Simulation software;
 - iv. Manager - Personnel whose main tasks is to manage personnel required to use or program simulation;
 - v. User - Personnel using simulation to experiment, test, train, etc...; and
 - vi. Maintenance Technician - Personnel required to maintain a scenario or simulation system.
- b. Programmer is described as follows:
 - i. Application Programmer - Personnel responsible to modify or create a simulation software;
 - ii. GUI Programmer - Personnel responsible to modify or create a simulation user interface software;
 - iii. CGE Programmer - Personnel responsible to modify or create Computer Generator Entities using a simulation software;
 - iv. RTI Programmer - Personnel responsible to modify/create interfaces with RTIs;
 - v. Control System Programmer - Personnel responsible to generate the code for control system using simulation software;
 - vi. Software Modeler - Personnel responsible to generate models and database;
 - vii. Scientist, Engineers and SME - Personnel responsible to model a system or system behaviour for a simulation.

- c. System Designer is described as follows:
 - i. Simulation System Designer - Personnel responsible to design a simulation system and identify the resources required;
 - ii. Control System Designer - Personnel responsible to design a control system and identify the resources required; and
 - iii. Scientist, Engineers and SME - Personnel responsible to design system(s) or system(s) behaviour for a simulation and identify the resources required.
- d. Visual Modeler is described as follows:
 - i. 3D Modeler - Personnel responsible to create 3D model to attach to simulation and simulated object; and
 - ii. Terrain Modeler - Personnel responsible to create the terrain for a simulation.

10.2.3 Recommended Educational Courses

The recommended Educational courses would be as follows:

- a. M&S/SE and Vendor's Simulation Software Starter;
- b. C++ for non-C++ programmer;
- c. C++ in Real Time for C++ Programmer;
- d. Debugging Windows-based Applications;
- e. Windows Programming using MFC – Visual Studio;
- f. Expert Systems;
- g. Modeling using MathLab/Simulink;
- h. 3D Visual Modeling;
- i. Terrain Modeling;
- j. 3D Visual and Terrain Modeling;
- k. Object-Oriented Design Pattern;
- l. Object-Oriented Analysis and Design;
- m. Introduction to Unified Modeling Language (UML);
- n. Applied Unified Modeling Language (UML);
- o. Modeling and Simulation Principles;
- p. Applied Concepts - Modeling and Simulation Principles;
- q. RPR-FOM v1.0 or v2.0: Overview and Specifications;
- r. Introduction to High Level Architecture (HLA);
- s. High Level Architecture (HLA) Specifications;
- t. FEDEP Process;
- u. Applied FEDEP;
- v. HLA Verification, Validation and Accreditation;
- w. Run-time Infrastructure (RTI),
- x. Introduction to Simulation Management;
- y. Simulation Management;
- z. Networked/Distributed M&S/SE;
- aa. Model Repository Management; and
- bb. Software Engineering/QA Applied to M&S/SE.

10.2.4 Recommended Training Courses

In addition, there is a requirement for recommended Training courses that would be specific to a Vendor's M&S/SE framework goods, as follows:

- a. M&S/SE framework Architecture;
- b. RTI Technologies;
- c. Distributed HLA Application;
- d. Simulation Runtime Services Layer;
- e. Common Simulation Services (CSS);
- f. Common Synthetic Environment Application;
- g. Simple Client Applications;
- h. Complex Client Applications;
- i. Server Applications;
- j. Observer/Participant Familiarization;
- k. Management Applications;
- l. Simulation Runtime Software Development Environments;
- m. Common Simulation Services Software Development Environments;
- n. Distributed HLA Application Software Development Environments;
- o. Client Application Software Development Environments;
- p. Server Application Software Development Environments and
- q. Specialized Training.

10.3 Professional Services

A DND-wide interoperable capability to simulate could only really take shape through Professional Services from Canadian Industries as the Teams, Groups and Sections from any Government Department are simply too small and too mobile. The Professional Services (PS) should provide a complete suite of services for the development, planning, integration, testing, implementation, operation, and migration of the Vendor's Suite of Goods to support DND's applications requirements.

These Professional Services are pre and post sales services that include Technical, Engineering, Management, and Consulting services. These resources can be leveraged for project intervention or long-term assignments related with the Goods and Services offered to DND.

10.3.1 Professional Services Resources Required

The Professional Services resources required would be in the following areas:

- a. Management/Consulting/Specialized Resources;
- b. Definition and Development Resources;
- c. Infrastructure, Operations and Maintenance Resources; and

- d. Program/Project Resources.

10.3.2 General Professional Services Categories Required

General Professional Services Categories required should be as follows:

- a. Analysis/Definition/Concept Development Support;
- b. Scientific/Engineering Support;
- c. Program/Project Management;
- d. Installation/Technical Support;
- e. Infrastructure Support;
- f. Integrated Logistic Support/LSA; and
- g. Administrative Support.

10.3.3 Tiered Professional Services Categories Required

Tiered Professional Services Categories required should include the following:

- a. Category 1 - Staff-Level (1–3 years experience) as follows:
 - i. Staff Engineers/Researchers with knowledge and experience in:
 - 1) Visual model integration;
 - 2) FOM editing/merging;
 - 3) Model / Entity creation and testing;
 - 4) Database population and testing;
 - 5) Scenario importation/migration between versions;
 - 6) Source code migration between versions;
 - 7) Supporting API usage with troubleshooting;
 - 8) GUI Implementation; and
 - 9) Programming.
 - ii. Technical Support Staff with knowledge and experience in:
 - 1) Supporting installation;
 - 2) First level troubleshooting;
 - 3) System Maintenance; and
 - 4) Network Maintenance.
 - iii. Administrative Support Staff with knowledge and experience in:
 - 1) Documentation;
 - 2) Data Entry; and
 - 3) Assets/Inventory Management.
- b. Category 2 - Mid (Operational)-Level (4–7 years experience) as follows:
 - i. Senior Technical staff/Group Leaders with knowledge and experience in:
 - 1) Lead Technical/Engineering Activity; and
 - 2) Software Engineering/QA.
 - ii. Project/System Managers with knowledge and experience in:
 - 1) Management of simple Projects;
 - 2) Software Engineering Project Management.

- iii. Subject Matter Specialists with knowledge and experience in:
 - 1) Application of System Requirements/Specifications; and
 - 2) Conduct of System Testing
- iv. Project Scientists/Engineers with knowledge and experience in:
 - 1) Basic Experimental design and data collection;
 - 2) Migration of complex components from other environments (software designer);
 - 3) Designing a generalized application programming interface (API) (software designer);
 - 4) Designing new components (software designer) as follows:
 - a. Framework;
 - b. Application;
 - c. Graphical User Interface (GUI);
 - d. Protocol;
 - e. Mathematical/Physics based models; and
 - f. Scheduling.
 - 5) Porting to new operating systems or variants (software designer); and
 - 6) Human Machine Interface definition and design as follows:
 - a. Menu/Display content;
 - b. Machine Hardware ergonomics; and
 - c. Basic User Interface Analysis.
- c. Category 3 - Senior-Level (8–14 years experience) as follows:
 - i. Manager Level Executives with knowledge and experience in:
 - 1) Management of facilities/networks;
 - 2) Consultation on infrastructure development; and
 - 3) Definition/Development of Project/Program Teams.
 - ii. Program/System Managers with knowledge and experience in:
 - 1) Management of complex Programs.
 - iii. Subject Matter Experts (SME) with knowledge and experience in:
 - 1) Development of System Requirements/Specifications;
 - 2) System Test Development; and
 - 3) Validation of Model Applications.
 - iv. Architecture Specialists with knowledge and experience in:
 - 1) Design of specialized system architectures.
 - v. Senior Scientists/Lead Engineers with knowledge and experience in:
 - 1) Definition, Design, Implementation of integrated Research Programs;
 - 2) Definition of Requirements;
 - 3) Approach for Model Development;
 - 4) Database and Tool Development;
 - 5) Methodologies for Migration/Conversion;
 - 6) Systems/Infrastructure Analysis;
 - 7) Definition of Processes for systems development and VV&A;
 - 8) Definition of Program Plans, Test Plans, Technical Specifications, and Statements of Work (SOW);

- 9) Leadership of Technical/Engineering Activity;
- 10) Knowledge Management; and
- 11) Total Information Awareness.

d. Category 4 - Expert-Level (15 and above years experience) as follows:

- i. Senior Level Executives with knowledge and experience in:
 - 1) Management of major programs;
 - 2) Management of major facilities/networks;
 - 3) Management Consulting on major infrastructure concepts; and
 - 4) Definition/Development of major Program Teams.
- ii. Senior Subject Matter Experts (SME) with knowledge and experience in:
 - 1) Development of Concept of Operations;
 - 2) Applications Requirements Analysis; and
 - 3) Operational Analysis.
- iii. Chief Scientists with knowledge and experience in:
 - 1) Applications Requirements Analysis;
 - 2) Scoping Studies; and
 - 3) Advanced Research Program Definition and Implementation
- iv. System Architects with knowledge and experience in:
 - 1) Designing specialized systems architecture;
 - 2) Designing systems of systems architecture; and
 - 3) Complex Network/Systems/Infrastructure Integration Planning.

11. Conclusion

A modular Modeling & Simulation/Synthetic Environment (M&S/SE) framework, with its associated Services, for developing and supporting a network-centric or distributed Collaborative Synthetic Environments (CSE) is proposed to promote, foster, augment, and expedite the standardization, interoperability, commonality, reusability, and seamless integration of legacy systems of M&S/SE in DND, Other Government Departments (OGD) and beyond. The modular M&S/SE framework relies on a network for communication between the various applications and legacy systems adapted to the framework. The M&S/SE framework, to develop and support distributed CSE, depends on a layered, functionally separated approach to building dynamically reconfigurable applications. Each layer of the framework provides successive levels of specialization so that as new technology evolves, the implementation of the layer can be changed to accommodate new hardware or technology changes. Together with the requirements for related Services, this Technical Memorandum has documented the network-centric M&S/SE framework requirements for an optimally interoperable, common and reusable distributed CSE in DND, and beyond, directly supporting Network-Centric Capability Management. It is recommended that the general segmentation of a modular M&S/SE framework should be as follows:

- a. Framework;
- b. Simulation Runtime;
- c. Software Development Environments (SDE);
- d. Client Applications;
- e. Server Applications;
- f. Distributed HLA Applications;
- g. Management Applications;
- h. Common Synthetic Environment (CSE) Infrastructure; and
- i. Dynamic Synthetic Environments/Computer Generated Forces (DSE/CGF).

In the context Capability management, DND has realized that a key transformational tool that will help the Department to procure or to deploy better existing and future Capabilities, faster and cheaper, is M&S/SE: ADM(Mat) is leading the DND with a Joint SMARTS vision of M&S/SE used at the enterprise level to better manage capabilities. DND is also realizing that it is costly and practically impossible to track the current ad hoc use of the M&S/SE Goods & Services throughout the department in order to provide or enhance interoperability and to avoid unnecessary duplication, incompatibility and redundancy. Therefore, DND must come up with innovative ways to overcome this problematic situation. One possible way is for DND to seek to get as many as possible applications that are already integrated to a simulation platform from either the vendor and/or its associated value-added partners to meet its solutions as long as the platform would also allow for third parties to integrate their products (open architecture concept); thereby stimulating follow-on business in Canada from smaller value-added companies.

The integration is the key value in terms of the current best practices. From decades of hard-lessons learned, by not working efficiently, have shown that an ad-hoc mixture of

interconnected services and components usually fails to work. Change will be difficult. Big changes will be more difficult. The adoption of distributed CSE will involve significant changes in how DND organizes duties and responsibilities of individuals, sections, and departments. Individuals, sections, and divisions will need to adopt new attitudes, accept more responsibility, learn new skills, master new approaches, and operate new systems - all in a faster-paced environment.

DND is now entering a period where it will not know the answer or the solution at the start of the process, and the techniques and tools that are currently associated with education and training may no longer be valid. This is where a network-centric CSE would come to play a crucial role in shaping tomorrow DND's decisions, by being used early from Concept Development & Experimentation all the way through Mission Rehearsal. Indeed, a network-centric CSE means a better linking of tools, a change of mindset, a better linking between coalition nations, a better linking within a nation (for public security for instance) and better linking within DND in terms of Network-Centric Capability Management.

References

1. Naisbitt, J. Megatrends. Warner Books, 290 pages, 1982.
2. Thompson, Marcel and Richard G. Brown. FFSE Section Contribution to the DRDC M&S/SE Cookbook in Support of the CASE Project, 9 February 2004
3. Brown, R.G., Marcel Thompson, and Claude Brochu. Software to Support Modeling Simulation in the Future Forces Synthetic Environments (FFSE) Section. R, December 2002
4. Thompson, M. Joint Simulation Network (JSimNet) and Modeling & Simulation Resources Repository Network (MSRRNet) Descriptions and Roles, and their Interactions, DRDC Ottawa Contractor Report, 1 October 2002
5. Thompson, M. A review of Defence Simulations and Strategies for the Future Forces Synthetic Environments (FFSE) Section. DRDC Ottawa Contractor Report, 26 October 2001
6. Thompson, Marcel, Eric Anschuetz, John Evans, JR Smullen, Gary Fraas, Arman Tchoubineh, Dan Paterson, Lucien Zalzman, Peter Ryan, Steve Wallo, Pete Butt, and Gary Estep. Key Network Infrastructure Issues for Coalition Distributed Simulation Experiments. TTCP Report TTCP AER TP-1, CP2B, June 2001
7. Alberts , David S., John J. Garstka, and Frederick P. Stein. Network Centric Warfare: Developing and Leveraging Information Superiority 2nd Edition (revised)., July 2002
8. Thompson, Marcel. DND Synthetic Environments Coordination Office (SECO) Strategic Plan. DRDC Ottawa Contractor Report, 26 June 2002.
9. Giles, L. Sun Tzu on the Art of War. <http://www.kimsoft.com/polwar.htm>, Last modified: 02/21/2003.
10. Vallerand, A.L. S. Albert and M. Thompson. Canadian Common SE framework Architecture. Proceedings of the SMi Inc Workshop on Virtual battlespace, London UK, Mar 2003, pp39-46.

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.) Defence R&D Canada – Ottawa Ottawa ON K1A 0Z4		2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable) UNCLASSIFIED	
3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.) Network-Centric Environment: A modular Modeling and Simulation/Synthetic Environment (M&S/SE) framework (U) .			
4. AUTHORS (Last name, first name, middle initial) Vallerand, A.L., Thompson, M.			
5. DATE OF PUBLICATION (month and year of publication of document) November 2004		6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.) 93	6b. NO. OF REFS (total cited in document) 10
7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.) Technical Memorandum. Final report			
8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.) DRDC Ottawa			
9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)		9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)	
10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.) DRDC Ottawa TM 2004-221		10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)	
11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification) <input checked="" type="checkbox"/> (X) Unlimited distribution <input type="checkbox"/> () Distribution limited to defence departments and defence contractors; further distribution only as approved <input type="checkbox"/> () Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved <input type="checkbox"/> () Distribution limited to government departments and agencies; further distribution only as approved <input type="checkbox"/> () Distribution limited to defence departments; further distribution only as approved <input type="checkbox"/> () Other (please specify):			
12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.) Unlimited			

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

DCD03 2/06/87

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

A modular Modeling & Simulation/Synthetic Environment (M&S/SE) framework for developing and supporting a network-centric or distributed Collaborative Synthetic Environments (CSE) is proposed and its specific and detailed requirements are documented here. It is proposed that such a framework is specifically designed to promote, foster, augment, and expedite the standardization, interoperability, commonality, reusability, and seamless integration of M&S/SE systems including legacy systems in DND, Other Government Departments (OGD) and beyond. The modular M&S/SE framework relies on a network for communication between the various applications and legacy systems adapted to the framework. To develop and to support a distributed CSE, the M&S/SE framework depends on a layered, functionally separated approach to building dynamically reconfigurable applications. Each layer of the framework provides successive levels of specialization so that as new technology evolves, the implementation of the layer can be easily changed to accommodate new hardware/software or technology changes. Together with the requirements for related and necessary Support Services, this Technical Memorandum documents in a logical approach the Network-Centric M&S/SE framework requirements for an optimally interoperable, common and reusable distributed CSE in DND and beyond, directly supporting Network-Centric Capability Management (U).

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

pen architecture framework; virtual environment; HLA; DIS; international standards; protocols; interoperability; reusability; commonality; scalability; distributed simulation; open programming structure; cross-platform supportability;

Defence R&D Canada

Canada's leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale



www.drdc-rddc.gc.ca